Model of a Proposed Local Interaction Mechanism Leading to Symmetric Global Structure in Virus Capsids

by

George E. Homsy
B.S.E. Electrical Engineering and Computer Science
University of California at Berkeley, 1987

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science
at the
Massachusetts Institute of Technology

September, 1997

Author: _____

George E. Homsy
Department of Electrical Engineering and Computer Science
September, 1997

Certified by: _____

Bonnie A. Berger
Associate Professor of Mathematics
Thesis Supervisor

Accepted by: _____

Arthur C. Smith
Chairman, Departmental Committee on Graduate Theses

Model of a Proposed Local Interaction Mechanism Leading to Symmetric Global Structure in Virus Capsids

by

George E. Homsy

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science
at the
Massachusetts Institute of Technology

September, 1997

ABSTRACT
A large class of biological viruses, the *icosahedral viruses*, have a protein capsid consisting of multiple coat protein monomers, bound together in a spherical lattice with icosahedral symmetry. In a majority of these icosahedral viruses, the coat protein monomers are all translated from the same gene, hence (presumably initially) identical in tertiary structure. Yet in the complete capsid they occupy slightly different binding environments according to their position. Moreover, the capsid is self-assembling. A completely satisfactory resolution of this apparent paradox has yet to be found. In this work, a novel pairwise interaction model is proposed that may explain how such capsids are able to self-assemble from their constituent monomers into icosahedrally symmetric large-scale structures. The mathematical construction of the model in parametric form is given; the construction of a numerical simulator to simulate the behavior of the model is outlined; and simulation results are presented and discussed.

Thesis Supervisor: Bonnie A. Berger
Title: Associate Professor of Mathematics

Acknowledgements

This thesis describes joint work with Professor Bonnie A. Berger, in the Department of Mathematics and the Laboratory for Computer Science at the Massachusetts Institute of Technology. It is an extension of, and is based on, earlier work by Bonnie Berger, Doug Muir, Peter Shor, and Russell Schwartz. The simulator and graphics routines described herein are based, with thanks, on code originally developed by them.

Thanks to Bonnie Berger for her advice and assistance.

Thanks to Pamela Thuman-Commike for supplying the micrographic reconstruction I used as the basis of figure 2.2.

Thanks to Peter Prevelige, Jonathan King, David Coombs, Adam Zlotnick, and Pamela Thuman-Commike, for valuable discussions about virology, and for their support and interest.

Thanks to Russell Schwartz, for discussions about simulation techniques and statistical mechanics.

Support

This work was supported by a Graduate Research Fellowship from the National Science Foundation. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the National Science Foundation.

Notes

A preliminary version of this work appeared at the Fifteenth Biennial Phage/Virus Assembly Meeting, held at Asilomar Conference Center in Monterey, California, from June 15th through June 20th, 1997.

Biographical Note

George Homsy took his Bachelor's of Science and Engineering in Electrical Engineering and Computer Science at U.C. Berkeley in 1987. Since then, he has worked in the fields of medical instrumentation design, operating systems research, and electronic CAD software design, all in San Francisco and the surrounding areas. Concurrently, he worked with several San Francisco based machine performance troups including Survival Research Laboratories and Amorphic Robot Works. After skidding off the yuppie track in 1992 to commune with his inner slacker, he expatriated himself to the Netherlands to pursue machine art and interactive electromechanical and pyroacoustic music performance. After two years, he became emotionally ready to pursue an advanced degree, and realized that higher education is one of the few fields in which the United States still leads the world. So he repatriated himself in Boston, where he remains to this day, enjoying research and the academic climate at MIT, and decrying the lack of wilderness and lonely deserted beaches.

Contents

List of Figures

Chapter 1

Introduction

1.1 Virus Structure

Many biological viruses consist of a genomic *core*, surrounded by a *capsid*, along with a *portal*, which serves as the virus's point of attachment to and entry into a host cell. Many plant and most human viruses fall into the class of *icosahedral* viruses, so named because the capsid, when observed using electron microscopy, displays two-fold, three-fold, and five-fold rotational symmetries. Hence it has the symmetry group of the regular icosahedron.

Many icosahedral capsids are generally composed of identical protein subunits, called *monomers*, bound together by protein-protein interactions to form complete, symmetric, and stable shells. In certain cases such capsids have been shown to be capable of self-assembly *in vitro* to form this stable configuration. This is amazing, since the monomers are translated from the same gene, and hence according to the "one gene, one protein" hypothesis, must be presumed identical in primary structure. But the tertiary structure of the monomers in a complete capsid is clearly not identical, since the capsid has sixty-fold icosahedral symmetry, but there can be many

times this number of monomers in a capsid. In this case, according to the "quasi-equivalence theory" of Caspar and Klug [5], the individual monomers are said to be in distinct and slightly different "binding environments" according to their position in the capsid, modulo the icosahedral symmetry group.

We are thus presented with an interesting problem: How do monomers, identical in sequence and hence (presumably initially) in tertiary structure, come to have distinct tertiary structures; and by what means do they spontaneously assemble into a configuration of icosahedral symmetry, when their initial spatial distribution and conformations do not display such symmetry?

1.2 Previous Work

Initial attempts at explaining capsid formation were largely phenomenological in nature: Caspar and Klug's [5] proposed "quasi-equivalence theory" of capsid formation was basically a statement that the subdivided icosahedral tilings found in virus capsids is a class of tilings which minimizes the average "deviation" of the tile shapes from some "nominal" shape.

Similarly, Tarnai, Gaspar, and Szalai [25] demonstrate that the packings of pentamers experimentally observed in "all-pentamer" virus capsids represent locally optimal packings of pentagons on the surface of a sphere. This is shown by starting with small spherical pentagons, constrained *a priori* to the surface of a sphere and initially not in contact, and "growing" them slowly until some contact occurs. They are then allowed to slide along each other, still growing, until they make more contacts and their corners are completely constrained. At this point, the packing is considered to be locally optimal.

Marzec and Day [15] discuss pattern formation in a more general class of spheroidal capsids, also from the standpoint of local optimization. But instead of rigid pentagons,

they use "morphological units", which are in effect spherical surface charge distributions with the desired symmetry. They define an "interaction energy", Σ^2, the surface integral of the sum of all pairwise products of the surface charge density. They then minimize this interaction energy by standard gradient descent techniques. The local minima so obtained closely reflect experimentally observed packings of monomers in certain capsids.

But these explanations are all of a descriptive form, relating the structure of complete capsids to the optimization of some parameter; little is said in these studies about possible pathways of assembly. By contrast, there has been more recent work on a "local rule based theory" of capsid assembly, which focuses on more operational descriptions.

Berger *et al.* [4] construct an operational model of capsid assembly capable of explaining the specificity of the resulting icosahedral structure, based on the assumptions that the monomers are capable of assuming distinct conformations, and that only interactions between sets of monomers in specific conformations are "allowed". In this model, the geometry of each conformation is specified as input to the model, along with a set of "binding rules", specifying all the allowable bonds between different binding sites on different conformations of monomers.

A numerical simulator has been constructed [17, 4] implementing this model. Experiments with the simulator have met with considerable success, both in explaining the formation of the complete capsid structures, and in successfully predicting certain types of malformations and modifications of structure which have been experimentally observed [1].

This work has been more recently extended by Schwartz *et al.* [23, 1, 24], who constructed a simulator which more realistically modeled the kinetics of capsid assembly, by explicitly modeling the time behavior of all monomers in a solution.

Although these models are operational at the level of the complete structure, they do not directly address the question of what type of mechanism might mediate the

protein-protein interaction, giving rise to the local rules for inter-monomer interactions. The rules are simply given in the model and not explained. This work attempts to fill in the details of the Berger-Shor model, by proposing a physical basis on which such local rules might be enforced.

1.3 The Tangential Flexibility Model

In this dissertation, I develop a model of capsid assembly which explains in operational terms how and why icosahedral capsids may develop *in vivo*. Since auto-assembly of capsids is a chemical process, however complex, any good model should explain capsid formation in both energetic and kinetic terms. That is, it should explain both *why* the complete capsid is the most energetically favorable structure, and *how* the structure is formed.

I address the first question by proposing a particular "energy model" which I call the Tangential Flexibility Model. By energy model, I mean a function mapping protein complexes made up of monomers, to energies. The TFM is a member of a class of models which I will call "high level ball and stick" models: A monomer is modeled as a collection of balls and sticks, with the balls representing entire domains of the protein (core, binding sites, *etc.*), and the sticks representing geometric relations between them. This type of model has been effectively used in prior studies of viral shell assembly [1, 2, 4, 23, 17] . The sticks may have springs attached between them to model the ability of the protein to flex in certain prescribed ways. The advantage of this model is apparent, in that it may accurately model nanometer-level physical behavior of a large protein, without resorting to detailed atomic level descriptions. This is appropriate, since if we wish to model protein-protein interactions in large complexes, we should be concerned not so much with the fine structure of the binding sites so much as their geometric relation to each other.

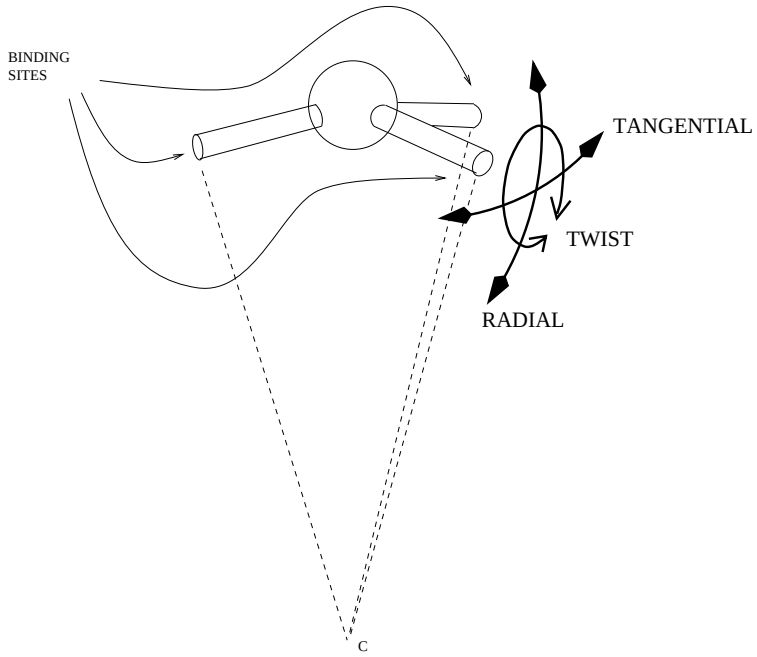

Figure 1.1: Diagram of a monomer in the TFM, showing binding sites, basis directions of flexibility, and ideal capsid centerpoint, C.

In more detail, the tangential flexibility model treats each capsid monomer as a dished triskelion, with a central core and three binding sites, as shown in figure 1.1. This approach is not new, having been used by [1, 2, 4, 23, 17] . The distinguishing feature of the TFM is that the three binding sites are more disposed to flexibility in the "tangential" direction: along the perimeter of the dish, than in the "radial" direction: perpendicular to the plane of the dish. Visual depictions of these basis directions are shown in 1.1. More detailed definitions are given in chapter 3.

The central idea behind this thesis is as follows: If a group of such monomers begins assembling according to their binding rules, their triskelion shape predisposes them to begin forming a hexagonal lattice on the surface of a sphere, with radius determined by the depth of the dish, as shown in figure 1.2. However, since the surface of a sphere cannot be regularly tiled with hexagons, the lattice is forced to start

deforming as it grows. If there were no preferential flexibility in the tangential direction, this would cause the edge of the lattice to become radially unstable, forming a saddle-like surface. But because of the preferential flexibility in the tangential direction, the hex lattice stays nearly constrained to the sphere as it deforms, and strain accumulates mainly in the tangential direction along the boundary of the growing lattice until a certain critical point at which the strain is so great that pentamers are introduced into the lattice. This point of critical strain is diagrammed schematically in figure 1.3.

The TFM can be described as a "cumulative strain" model, in that it seeks to explain the appearance of irregularities (i.e. pentamers) in the capsid structure as a result of strain accumulating in a growing structure. A potential problem with such cumulative strain models, however, is the existence of many metastable states with incorrect global topology for any given energy model. If we were to try a simple sequential simulation of assembly, such a metastable state could lead the assembly process down a path which does not lead to the global minimum energy. But since the dynamics of chemical assembly tends to seek out lower energy states by trying many possible states, it would seem that this shortcoming is only a problem for simulations with a simple sequential dynamics, and not a fundamental flaw in the underlying physical model. So I have used a more advanced simulation technique to study this model, as described below.

1.4 The Simulator

Given the number of accessible metastable states on the path to formation of a complete capsid, it is apparent that a simulator more sophisticated than a sequential assembly simulator is needed. The family of simulators used by [1, 2, 4, 17] are sequential simulators, with some qualifications: Although they have the capability of adding a new node to a random location instead of a deterministically, sequentially

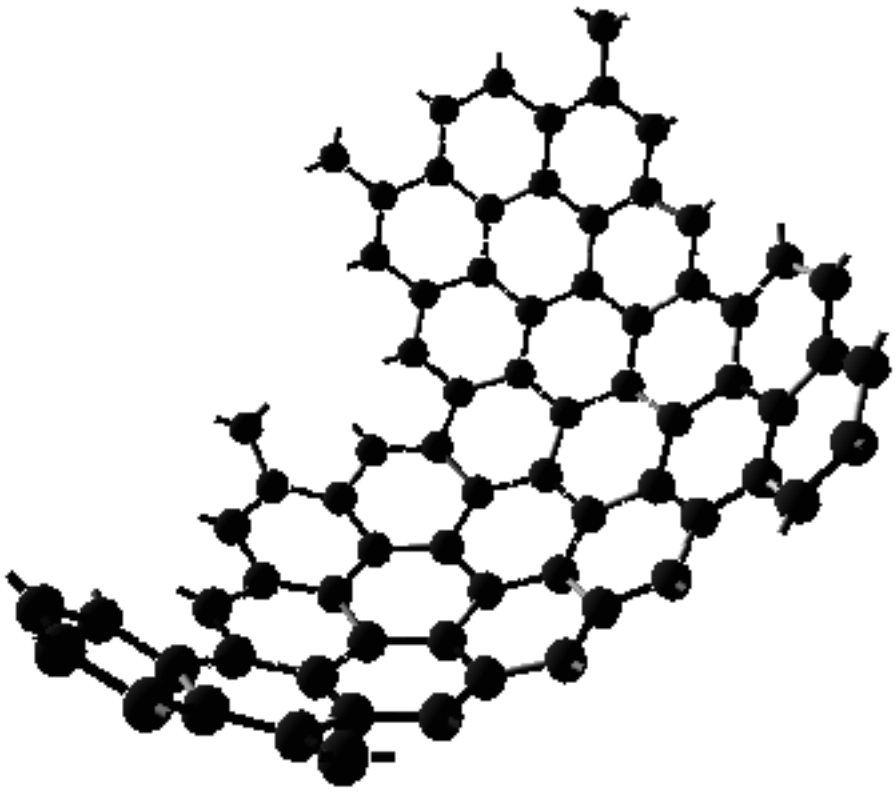

Figure 1.2: Some monomers assembling in a hexagonal lattice, constrained approximately to the surface of a sphere.

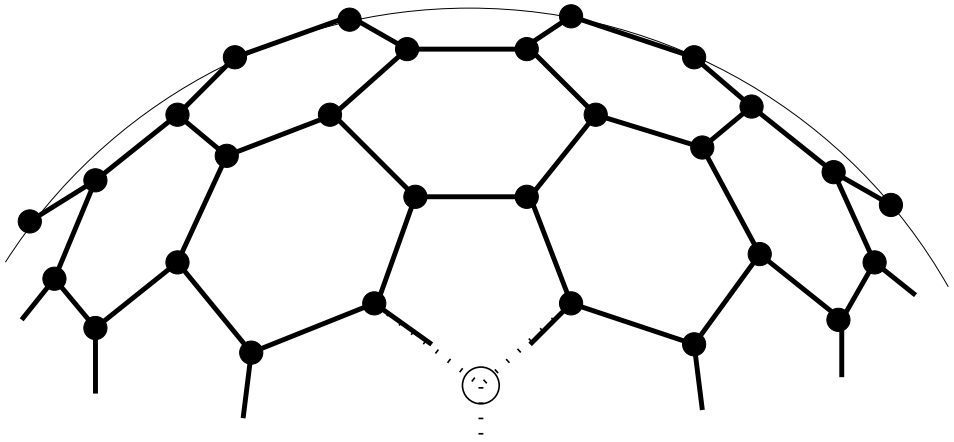

Figure 1.3: The hexamers have accumulated such strain at the edge of the growing lattice, that a pentamer is about to be introduced.

selected one, they have limited capability for backtracking on previously made assembly decisions. When the TFM is run on such a simulator, the simulations typically get stuck in a metastable state.

The simulator of Schwartz *et al.* [23, 24] deals with this problem at a very detailed level, by adopting a dynamical approach: The position and velocity of every monomer in solution is integrated dynamically, and the monomers all have different, thermally determined probabilities of binding or unbinding from a given site. This certainly solves the backtracking problem, but requires a great deal of computation.

So to more accurately model the final structures engendered by the TFM in a real chemical environment, but without the computational overhead necessitated by a dynamical simulation, I have modified the simulator of [1, 2, 4, 17] to support randomized assembly simulations using the Metropolis method. This is a well known Markov Chain Monte Carlo technique for running stepwise simulations on large thermodynamic systems which allows measurement of approximately correct thermodynamic averages without exhaustive enumeration of the space of possible states. This

is good, since the size of the state space (*i.e.,* the number of admissible topologies) is quite large. Due to the geometric irregularity of the problem, I have not been able to count the state space explicitly, but it is probably exponential in the number of monomers in a complete capsid.

Use of the Metropolis method affords at least two advantages in this context, both related to the large size of the state space:

- The probability distribution of the Markov Chain may converge to the equilibrium distribution in polynomial time. Hence, using the Markov Chain to find a state with approximately optimal energy may be faster than other optimization methods.

- More importantly for our knowledge of the biochemistry, since the moves used in the Markov Chain are "physically natural" ones, corresponding to discrete chemical events, we can argue that the mixing time of the chain is closely related to the actual assembly time of capsids. This is significant, since even if we could show by some other optimization technique that the optimal state of the TFM is in fact the correct capsid topology, that still would say nothing about how fast the capsid could form. Indeed, it is imaginable that at the rate at which chemical events are occurring, the large size of the state space might result in an optimal capsid not forming within any reasonable length of real time. The close correspondence between the moves of our Markov Chain and chemical events allows us to argue that the optimal capsid state is, in fact, reachable within a reasonable amount of real time in a real chemical environment.

1.5 Structure of this Dissertation

The structure of this dissertation largely mirrors the structure of the introduction.

Chapter 2 discusses current knowledge regarding structure and assembly of icosahedral virus capsids in more detail.

Chapter 3 discusses the TFM in detail, including a discussion of important steps in my thinking along the path to the current state of the theory.

Chapter 4 discusses the design of the simulator in detail, including how support for the TFM was added, how support for the Metropolis Algorithm was added, and how the user interface was improved to allow more in-depth investigations of the energetics of capsid assembly. Chapter 4 also contains a brief discussion of the implementation issues involved in the construction/modification of the simulator. However, since the TFM, not the simulator, is the main point of this dissertation, I have made every effort to keep this discussion brief.

Chapter 5 gives a discussion of Markov Chain Monte Carlo methods in general, and specifically introduces the Metropolis Method as an appropriate statistical mechanics algorithm by which to direct the progress of simulations. A brief discussion of the implementation of the Metropolis Method within the context of the TFM simulator is also given.

And finally, chapter 6 discusses several investigations I have made using this simulator, and the results of these. Connections with previously unexplained biological phenomena are pointed out, and avenues for further research, both with the simulator and in the laboratory, are suggested.

Chapter 2

Icosahedral Virus Capsids

2.1 Introduction

Encapsulated viruses (as opposed to filamentous viruses) generally consist, at the simplest level of description, of a nucleic acid genome, or "core", surrounded by a capsid. The capsid, made of protein subunits sometimes referred to as "coat proteins", serves to enclose, protect, and stabilize the genome. Many viruses also have a "portal complex", a complex of proteins which allows the virus to attack and fuse with a host cell. A highly schematized picture of a virus fusing with host cell is shown in figure 2.1 (a) and (b).

The viral genome, following fusion, either inserts itself into the host genome or simply floats in the cytoplasm. In any case, the cellular transcription and/or translational mechanisms of the host cell are "tricked" into producing replicas of the viral genome and the coat proteins (figure 2.1 (c)). The replicas so produced assemble into complete viruses, each one a copy of the original infecting virus (figure 2.1 (d)). Eventually so many daughter viruses are produced that the host cell becomes mechanically destabilized and lyses, releasing up to many thousand "daughter" viruses

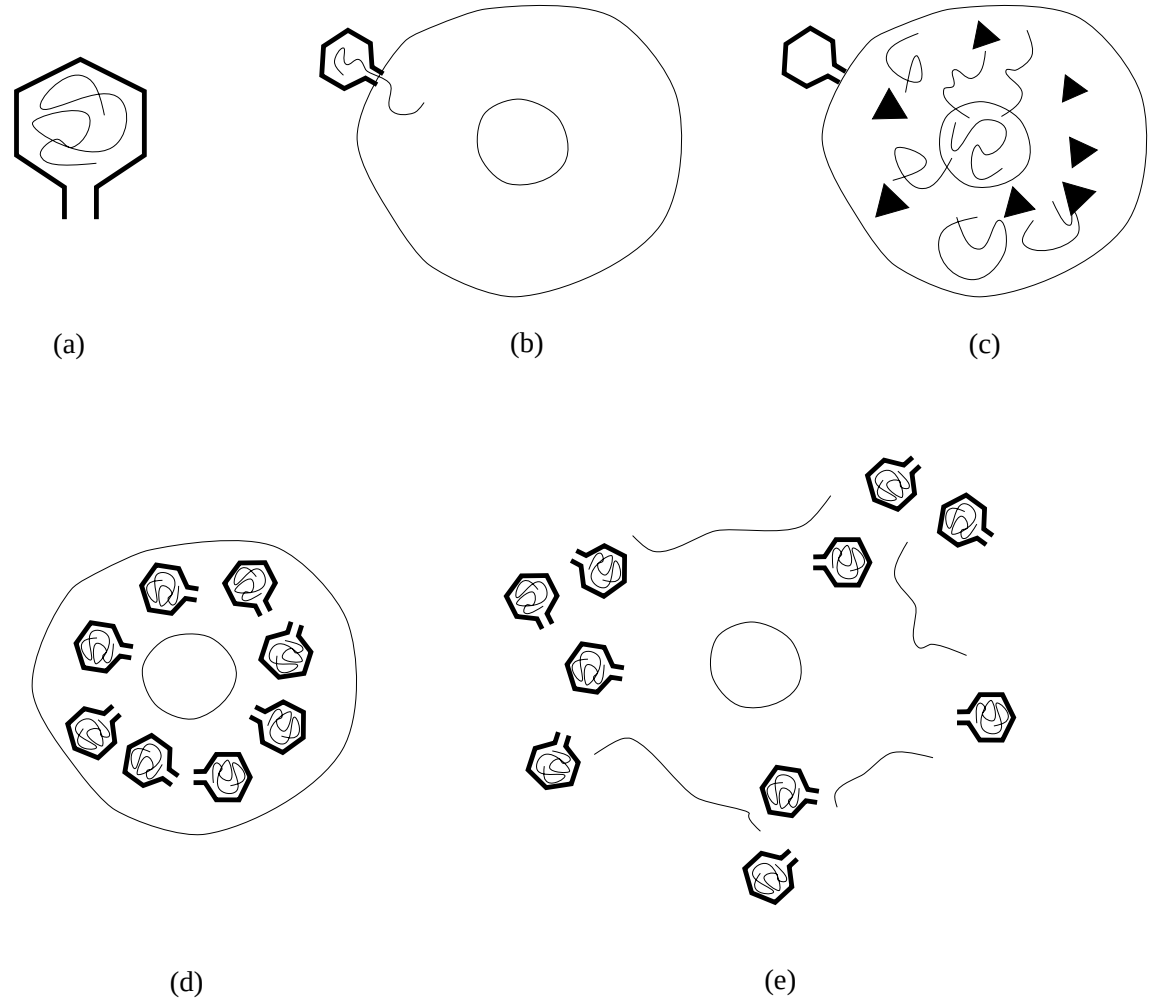

Figure 2.1: The "life" cycle of a virus, highly schematized.

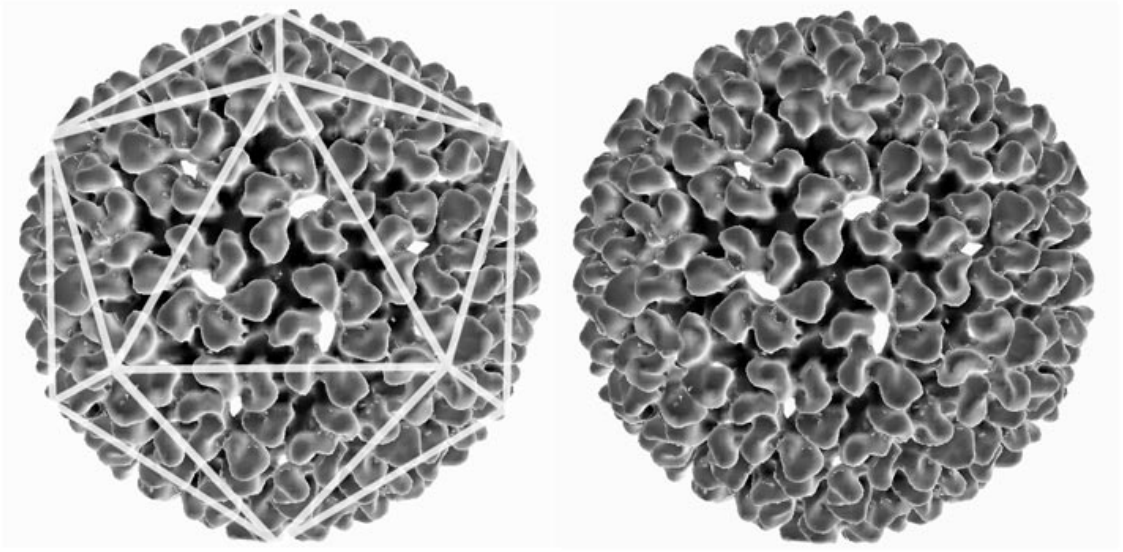

Figure 2.2: A micrographic reconstruction of an icosahedral virus capsid, with (a) and without (b) the icosahedral symmetry superimposed graphically. Reconstruction and image courtesy of Pamela Thuman-Commike.

into the intercellular space. The daughter viruses can then attack and fuse with neighboring host cells, thus starting the cycle anew (figure 2.1 (e)).

A large class of plant viruses, and most human and animal viruses, have a so-called "icosahedral capsid". That is, when one considers the capsid as an agglomeration of individual coat proteins, it has the same symmetry group as the icosahedron (see figure 2.2).

2.2 Capsid Structure of Quasi-Equivalent Viruses

A large subclass of the icosahedral viruses is the class of "quasi-equivalent" viruses, so named by Caspar and Klug [5]. These have the following properties:

- They have coat proteins which are all translated from the same gene, hence all equivalent in primary structure.

- The coat proteins appear nonetheless in slightly different "binding environments" in the finished capsid. That is, the neighbors of monomer A can bear different geometric relations to each other and to A than do the neighbors of monomer B.

- The monomers bind together in six-fold rings, called *hexamers*, in most places in the capsid except at the vertices of the icosahedron. At these points, the monomers are bound together in five-fold rings (called, appropriately enough, *pentamers*, or *pentons*). Hexamers and pentamers are collectively called capsomers. See figure 2.3.

- The binding environments, though not exactly the same, are similar in geometry, due to the high order (sixty-fold) symmetry of the capsid, and due to the topological similarity of always being bound together in five-fold and six-fold rings.

Berger *et al.* [1, 2, 4, 23, 17] were the first to formalize the binding relations between monomers in a form which permitted combinatorial analysis. In this model, the monomers of quasi-equivalent capsids are assumed to have three binding sites each, arranged in a triskelion, as shown in figure 2.4. They are chiral, and the binding sites are all different in character.

Now that we have defined our basic model of a monomer, we may define a notation for capsid topology. This again will be the same notation as used by Berger *et al.* [1, 2, 4, 23, 17] . Let us denote a single monomer by a circle with three edges emanating from it, representing binding sites, and let us label each binding site with an in-arrow, an out-arrow, or a dotted line, as shown in figure 2.5 (a). Let us call the binding sites "IN", "OUT", and "ODD", respectively. We then define binding rules as follows:

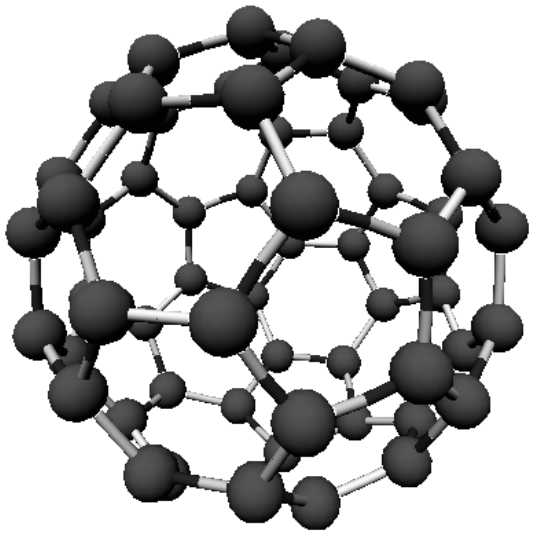

Figure 2.3: A schematic diagram of an icosahedral capsid. Note that the monomers are arranged in most places in rings of six, but in some places in rings of five.

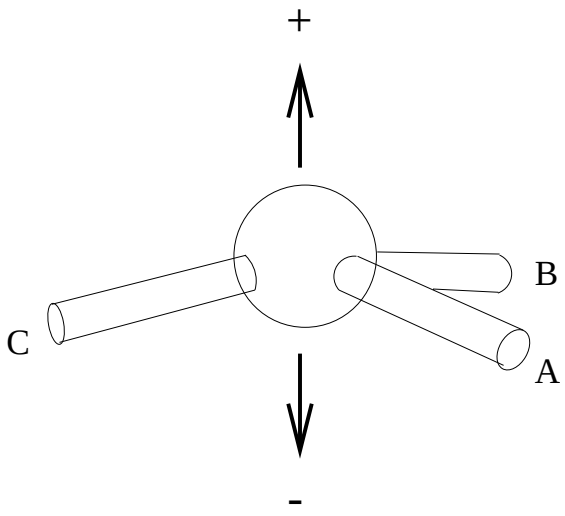

Figure 2.4: A cartoon representation of a single monomer, showing the different binding site types. The + and - signs explicitly indicate the monomer's chirality.

- IN may bind to OUT
- OUT may bind to IN
- ODD may bind to ODD

We may now draw schematic diagrams of topologies, an example of which is shown in figure 2.5 (b). This simple topology is a subtopology of a hexagonal lattice. Note that there are two types of hexamers: Those with all IN-OUT bonds, and those with alternating IN-OUT and ODD-ODD bonds. Let us call those with all IN-OUT bonds, *primary hexamers*, and let us call the others, *mixed bond hexamers*. Clearly, two primary hexamers cannot share an edge. In fact, a moment's reflection will show that primary hexamers must be separated by rings of mixed bond hexamers, and that this is the only type of hex-lattice which can be constructed with these binding rules.

A more instructive example is shown in figure 2.5 (c). This is a complete face of a T=3 capsid. The numbers inside the circles denote one possible assignment of

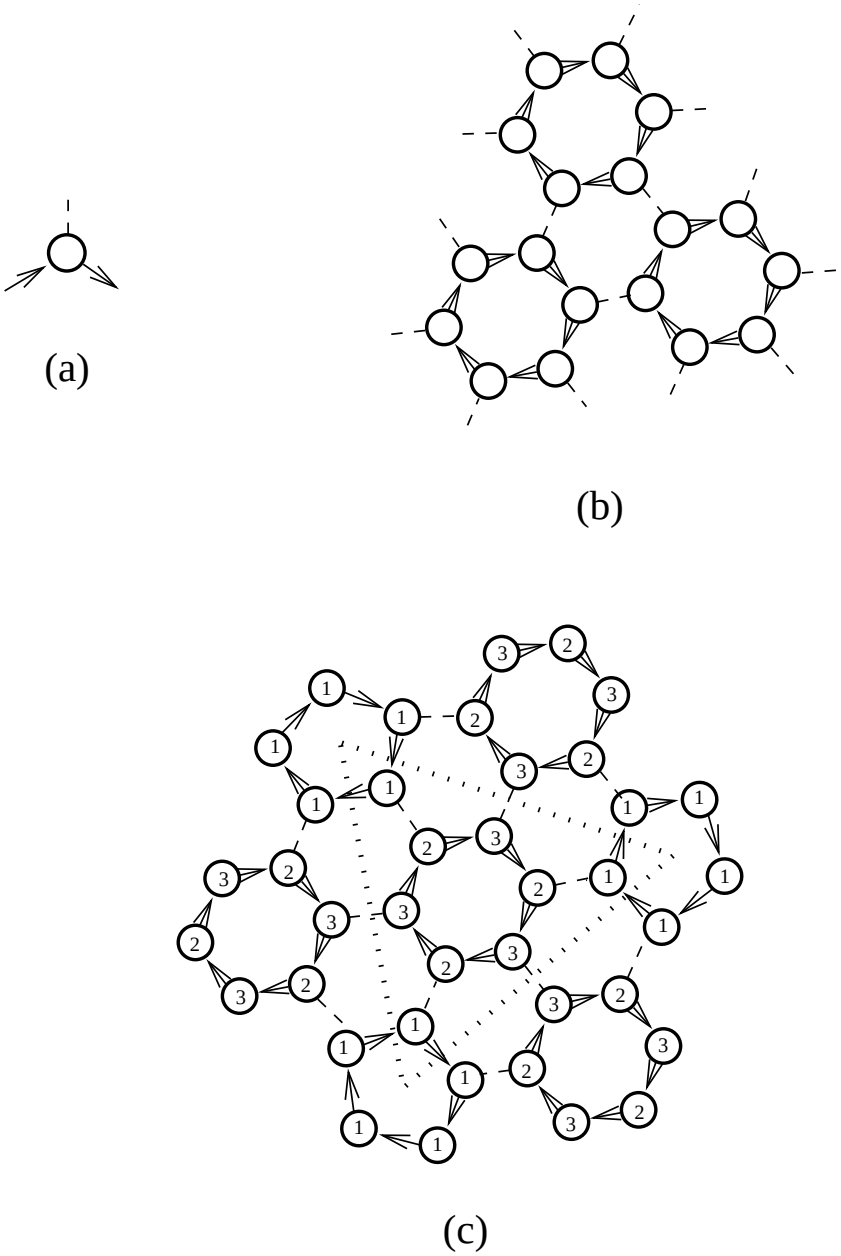

Figure 2.5: Schematized topologies: (a) A single monomer. (b) A small hex lattice. (c) One face of a T=3 capsid, with dotted lines describing the face boundary. The numbers indicate one possible assignment of Berger-Shor node types.

Berger-Shor types to the nodes (see [3]). Note that this is almost a sublattice of a hex lattice as before, but three of the primary hexamers have been replaced by pentamers. These pentamers lie at the vertices of the icosahedron which define the capsid symmetry.

2.2.1 A Counting Argument

The T-number of a quasi-equivalent capsid can be defined in two ways:

- Capsid-centric definition: The T-number is 1/60th the number of monomers making up the capsid: $N = 60T$.
- Monomer-centric definition: The T-number is the number of distinct binding environments (modulo the icosahedral symmetry group) in which monomers are found.

It has been noted [5] without proof that the T-number of a quasi-equivalent capsid is always of the form:

$$T = h^2 + hk + k^2, \tag{2.1}$$

where h and k are nonnegative integers. This was restated, and proved by Berger and Shor [3], using an area argument.

In refinement of this, and as a side note, I present here an alternative and perhaps simpler proof of the same fact, based on topology and counting:

The face of any icosahedral quasi-equivalent capsid is, by definition, of the form shown in figure 2.6 (a): An equilateral triangle, $\mathcal{T}$, (indicated by the dotted triangle) inscribed in a skew-symmetric hexagon H (indicated by the heavy black lines), as shown. The small triangles here represent monomers, with a bond denoted by the shared edge of two such small triangles. The circled points represent centers of pentamers, defining a face of the icosahedron.

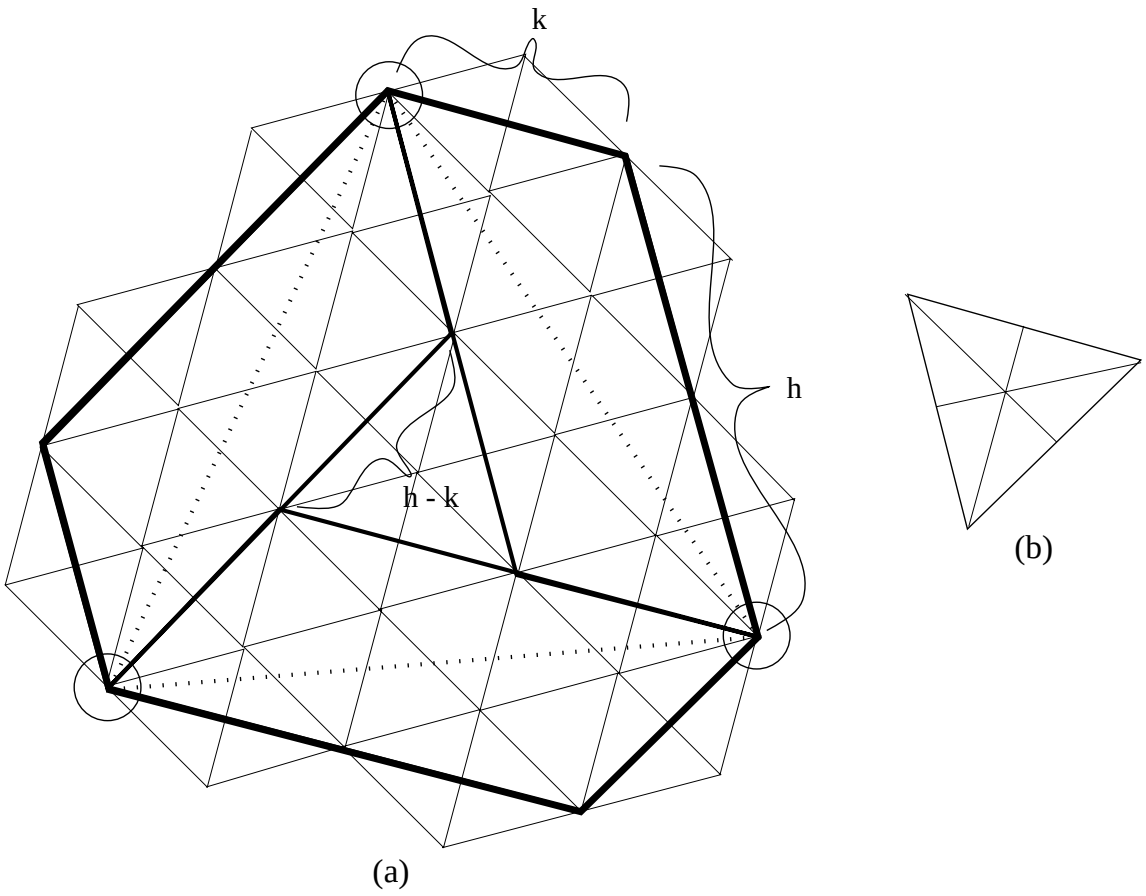

Figure 2.6: Diagram for the counting argument. $\mathcal{T}$ is denoted by a dotted line, H by a heavy black line.

There is a restriction that the sides of H be drawn perpendicular to the bond edges, and that they be of lengths which are multiples of two triangle altitudes. This restriction comes from the fact that, as we noted above, pentamers and primary hexamers are separated by rings of mixed bond hexamers, and H must have its vertices centered on pentamers or primary hexamers.

Let h and k be defined as shown. Clearly, h, k, and the handedness of H are sufficient to define $\mathcal{T}$. Let us now count the number of monomers in $\mathcal{T}$: First, define a type of triangle as shown in figure 2.6 (b), and denote it as a $\mathcal{T}'$ triangle. There are $2hk$ $\mathcal{T}'$ triangles per parallelogram, and there are three parallelograms, but only half of each parallelogram is in $\mathcal{T}$, so there are $\frac{3}{2} \cdot 2hk$ $\mathcal{T}'$ triangles in parallelograms. There are $(h-k)^2$ $\mathcal{T}'$ triangles in the central triangle. So, in total, there are $\frac{3}{2} \cdot 2hk + (h-k)^2 = h^2 + hk + k^2$ $\mathcal{T}'$ triangles in $\mathcal{T}$. Finally, there are 3 monomers per $\mathcal{T}'$ triangle, and there are 20 faces to the icosahedron, so the total number of monomers in the capsid is:

$$N = 20 \cdot 3(h^2 + hk + k^2) \equiv 60T. \tag{2.2}$$

This is the defining relation for T.

2.3 The Assembly Problem

As stated in the previous section, the capsid proteins are all transcribed from the same gene, hence they are all identical in primary structure. By the hypothesis that the primary structure determines the tertiary (*i.e.,* spatial) structure of a protein, we must suppose that all the capsid proteins are identical in tertiary structure, at least as long as they are free floating in the cytosol.

However, since the finished capsid displays distinct binding geometries for distinct capsid proteins, we know that the tertiary structure *cannot* be identical for bound capsid proteins.

If we now recall the fact that capsids assemble spontaneously in chemically favorable conditions, both *in vivo* and *in vitro*, we can write down, in summary form, the

Quasi-Equivalent Capsid Assembly Problem:

- Capsids are self-assembling;
- Monomers in capsids exist in distinct shapes, with global symmetry, *BUT*,
- Monomers have the same primary structure, thus are presumably indistinguishable in solution.

The question then is: *how does this happen?* As capsids self-assemble, monomers coming out of solution and assembling to the nascent capsid must somehow assume their distinct conformations at some point in time. They do so without direct external assistance, and they do so symmetrically, so as to result in a structure with global symmetry. This thesis proposes one possible explanation for how this might come about.

Chapter 3

The Tangential Flexibility Model

This chapter covers the Tangential Flexibility Model in detail.

3.1 Basic Structure of the TFM

The Tangential Flexibility Model, or TFM, is an "energy model", as introduced in chapter 1. That is, it is a mapping from states to real numbers. The real number associated with each state is interpreted as the "energy" of the state. The mapping from states to energies given by the model is used in simulation to attempt to arrive at an equilibrium state; that is, a state of low energy.

3.1.1 Definitions

A *monomer*, sometimes called a *node*, is a rigid structure in $\mathcal{R}^3$, with some additional parameters: It is composed of a *body*, with a position and orientation in $\mathcal{R}^3$, and an indexed set $\{B_j\}$ of *binding sites*. For purposes of definition, the body is assumed to be at the origin of a local coordinate system, with identity rotation, and each binding site B_j is given as a pair (V_j, R_j), where $V_j \in \mathcal{R}^3$ is interpreted as the position of

the binding site, and $R_j \in SO(3)$ is interpreted as the orientation of the binding site, both relative to the local coordinate system. Each monomer can move and rotate freely in $\mathcal{R}^3$, but its binding sites never alter their relative positions nor orientations. The ray from the body to a binding site is called an *edge*.

Each monomer has a *Berger-Shor type* associated with it, which is the type number of the monomer as used in the Berger-Shor local rules model [4]. Generally, in the TFM, the Berger-Shor type of all monomers is set to the same value, reflecting the fact that the TFM makes no assumptions about monomers coming in *a priori* distinct types.

Each binding site also has some associated parameters: A *neighbor type*, a *separation parameter*, and three *flexibility parameters*. The neighbor type specifies, in the Berger-Shor model, the type of node to which this binding site may connect. The separation parameter, k_{sep}, is the spring constant of a linear spring which is connected between connected binding sites in a *state* (see below). The energy of a state is increased by the energy stored in the spring, if connected binding sites do not exactly spatially coincide. See figure 3.1 (a).

The three *flexibility parameters* are also spring constants. These relate not to position, but to relative orientation of connected monomers:

- The *radial flexibility parameter*, k_{rad}, is interpreted as the spring constant of a (nonlinear) spring which tends to keep the two edges associated with connected binding sites aligned in the radial direction (see figure 3.1 (b)). For a more detailed discussion of why and in what manner this spring is nonlinear, see section 3.1.2.

- The *tangential flexibility parameter*, k_{tang}, is the rate constant of a spring which keeps connected edges aligned in the tangential direction (see figure 3.1 (c)).

- The *twist parameter*, k_{twist}, is the rate constant of a torsional spring which tries to keep two connected binding sites in their correct nominal orientation (see figure 3.1 (d)).

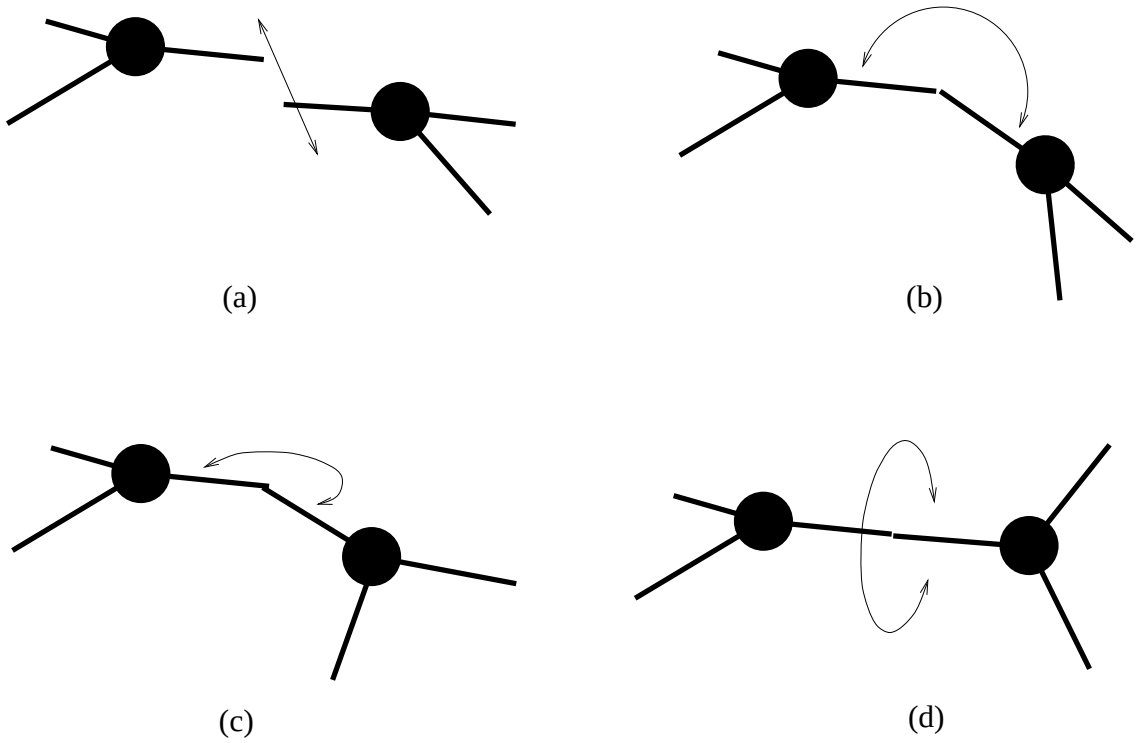

Figure 3.1: Degrees of freedom in binding site positioning. (a) Separation, (b) Radial, (c) Tangential, (d) Twist.

A *bond* is an unordered pair of binding sites, which are interpreted as being chemically connected by protein-protein interaction forces.

A *topology* is a collection of monomers, and a collection of bonds specified between them. Note that no positions or orientations of the monomers are specified: A topology is strictly a combinatorial entity.

Finally, a *state* is a topology, along with a specification of the exact positions and orientations of all monomers in $\mathcal{R}^3$. If the energy of a state has been minimized (see section 3.1.3), then the state is said to be *optimized*. Otherwise, the state is said to be *raw*.

3.1.2 Energy Model

The TFM is a *pairwise* energy model. That is, the energy of a state can be expressed as a sum over all (nodes and) bonds in the state: No three-way or higher interactions are included or allowed.

The energy of a state S in the TFM can be written as the sum:

$$E(S) = E_{gibbs}(S) + E_{bonds}(S) + E_{geom}(S). \tag{3.1}$$

Here, E_{gibbs} represents the decrease in Gibbs Free Energy associated with removal of the monomers from free solution and adding them to the state, E_{bonds} represents the energetic decrease associated with bond formation, and E_{geom} is the spring energy due to accumulated stress in S.

Since all the monomers are identical, and if we assume their concentration in solution does not change appreciably during capsid formation, we may write:

$$E_{gibbs}(S) = k_{gibbs}|V_S|, \tag{3.2}$$

where V_S denotes the node set of S.

If we further assume that the energy of bond formation does not depend on which bond is forming, we may write also:

$$E_{bonds} = k_{bond}|B_S|, \tag{3.3}$$

where B_S denotes the bond set of S. Note that, with this set of assumptions, E_{gibbs} and E_{bonds} depend only on the topology, not on the geometry, of S.

This leaves us with only the geometric energy E_{geom} left to specify. The first two energy terms, E_{gibbs} and E_{bonds}, are needed only to "drive" the capsid formation reaction forward chemically; they do not specify anything interesting about the final capsid structure. By contrast, E_{geom} is more complex than the other two; indeed, it is the heart of the TFM, since it results in selective flexibilities of bonds.

The geometric term for a pairwise energy model is written generally as a sum of bond geometric energies:

$$E_{geom}(S) = \sum_{b \in B_S} E_g(b). \tag{3.4}$$

If b is written as an ordered pair of binding sites: $b = (b^1, b^2)$, then its geometric energy $E_g(b)$ can be written:

$$E_g(b) = E_g(b^1, b^2) = f(X^1, O^1, X^2, O^2), \tag{3.5}$$

where X^i denotes the position vector of binding site b^i, and O^i denotes the orientation of b^i.

We would like to require that the function f has certain symmetries. In particular, we require f to be invariant under rigid transformations, since our choice of coordinate system is arbitrary; and we require that f be symmetric under the interchange of indices $1 \leftrightarrow 2$, since our numbering of nodes and binding sites is arbitrary. More formally:

- $f(X^1 + W, O^1, X^2 + W, O^2) = f(X^1, O^1, X^2, O^2)$ (translational invariance)

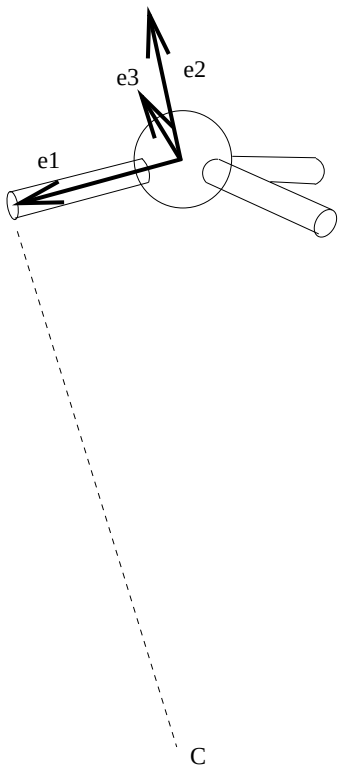

Figure 3.2: Definition of the unit vectors describing the orientation of a binding site.

- $f(RX^1, RO^1, RX^2, RO^2) = f(X^1, O^1, X^2, O^2)$, where R is a rotation (rotational invariance)
- $f(X^2, O^2, X^1, O^1) = f(X^1, O^1, X^2, O^2)$ (node indistinguishability; symmetry under interchange of indices)

From the translational invariance condition, we have that f cannot depend on X^1 and X^2 independently; it can only depend on the difference of X^1 and X^2.

Now, before proceeding further, let us specify a representation for the X^i's and the O^i's:

X^i, as mentioned in section 3.1.1, is the representation in $\mathcal{R}^3$ of the centerpoint of node i. Let us further represent O^i, the orientation of the binding site on node i,

by a 3×3 matrix, whose column vectors will represent the basis vectors of the local coordinate system of binding site b^i.

The e_1 column vector will be taken to represent the direction of protrusion of the binding arm from the centerpoint of the node. The e_2 column vector will be understood to represent the unit vector in the radial "outward" direction. This is only a nominal "outward" direction, since we are not yet considering a closed capsid here; it is the direction which *would* be outward if we had a complete capsid. And e_3, naturally, is chosen to define an orthonormal, right handed coordinate system: $e_3 = e_1 \times e_2$. This situation is diagrammed in figure 3.2.

Note that O is considered to be centered at the centerpoint of the node. Since e_1 points in the direction of the binding arm, this means that for a unit-length binding arm, the binding site is located at $X + e_1$. We may now define the *nominal* configuration of a bond: Two binding sites in "nominal" configuration have the following properties:

- Their binding arms are oppositely oriented;
- Their e_2 vectors are aligned; and
- Their positions are coincident.

This situation is shown in figure 3.3.

Now we wish to try to define the energy function E_g, as a quadratic form in the binding site separation and in the various deflection angles (as per section 3.1.1) as follows:

$$E_g(X^1, O^1, X^2, O^2) = \frac{1}{2}k_{sep}l_{sep}^2 + \frac{1}{2}k_{rad}\Theta_{rad}^2 + \frac{1}{2}k_{tang}\Theta_{tang}^2 + \frac{1}{2}k_{twist}\Theta_{twist}^2. \quad (3.6)$$

Here, l_{sep} is the vector representing the separation spring joining the binding sites, and the Θ's are the deflection angles (understood to be functions of X^1, O^1, X^2, O^2).

We must now define l_{sep} and the Θ's in terms of the X's and O's. If we ensure that each of these mappings is either symmetric or antisymmetric under the interchange of indices $1 \leftrightarrow 2$:

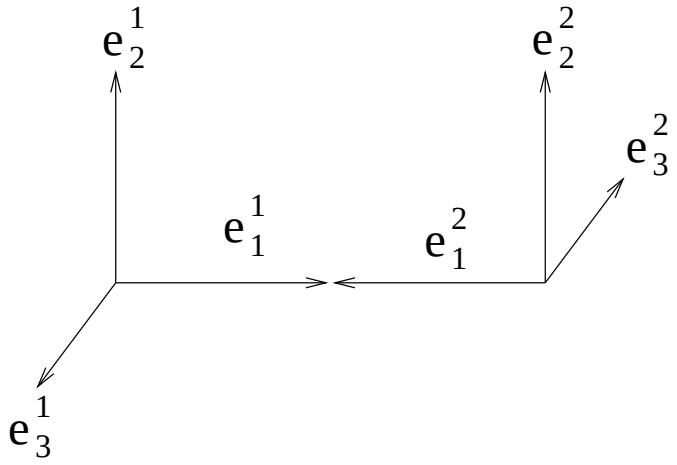

Figure 3.3: Nominal position of two bound sites, relative to each other.

$X^1 \leftrightarrow X^2$ $O^1 \leftrightarrow O^2$	$l_{sep} \leftrightarrow \pm l_{sep}$ $\Theta_{rad} \leftrightarrow \pm\Theta_{rad}$ $\Theta_{tang} \leftrightarrow \pm\Theta_{tang}$ $\Theta_{twist} \leftrightarrow \pm\Theta_{twist}$

then we will be assured that E_g will be symmetric under the interchange of indices, since it is a quadratic form.

Let us consider first Θ_{twist}. If b^1 and b^2 are otherwise in nominal configuration (that is, there is no radial or tangential bending), then e_1^1 and e_1^2 are collinear. In this case, the sine of the twist angle is given by:

$$\sin(\Theta_{twist}) = -e_2^1 \cdot e_3^2. \tag{3.7}$$

It is also given by:

$$\sin(\Theta_{twist}) = -e_2^2 \cdot e_3^1. \tag{3.8}$$

For small angles, $\sin(\Theta) \approx \Theta$, we may linearize these forms to first order by simply approximating:

$$\Theta_{twist} \approx -e_2^1 \cdot e_3^2, \tag{3.9}$$

$$\Theta_{twist} \approx -e_2^2 \cdot e_3^1. \tag{3.10}$$

To obtain a form which is more nearly insensitive to variations in the two bending angles Θ_{rad} and Θ_{tang}, it is reasonable to define Θ_{twist} as the mean of these two quantities:

$$\Theta_{twist} \equiv \frac{1}{2}(-e_2^1 \cdot e_3^2 - e_2^2 \cdot e_3^1). \tag{3.11}$$

Clearly, this definition has the required symmetry under interchange of indices, and it reduces to the exact twist angle, for small twists between binding sites with $\Theta_{rad} = \Theta_{tang} = 0$.

We proceed similarly for the other deflection angles, to obtain:

$$\Theta_{tang} \equiv \frac{1}{2}(-e_1^1 \cdot e_3^2 + e_2^2 \cdot e_3^1) \tag{3.12}$$

$$\Theta_{rad} \equiv \frac{1}{2}(+e_1^1 \cdot e_2^2 + e_1^2 \cdot e_2^1). \tag{3.13}$$

Note that Θ_{tang} is antisymmetric, whereas Θ_{rad} is symmetric. This makes perfect sense given the definition of nominal position, shown in figure 3.3. These definitions also both have the desired properties of reducing to the exact angle in question for small angles, with no interaction from other deflections.

Finally, l_{sep}, the separation vector, is a simple matter:

$$l_{sep} \equiv (X^2 + e_1^2) - (X^1 + e_1^1), \tag{3.14}$$

which is antisymmetric.

In summary, then, we have

$$E_g(X^1, O^1, X^2, O^2) = \frac{1}{2}k_{sep}l_{sep}^2 + \frac{1}{2}k_{rad}\Theta_{rad}^2 + \frac{1}{2}k_{tang}\Theta_{tang}^2 + \frac{1}{2}k_{twist}\Theta_{twist}^2, \tag{3.15}$$

where

$$l_{sep} = (X^2 + e_1^2) - (X^1 + e_1^1), \tag{3.16}$$

$$\Theta_{rad} = \frac{1}{2}(+e_1^1 \cdot e_2^2 + e_1^2 \cdot e_2^1), \tag{3.17}$$

$$\Theta_{tang} = \frac{1}{2}(-e_1^1 \cdot e_3^2 + e_2^2 \cdot e_3^1), \tag{3.18}$$

$$\Theta_{twist} = \frac{1}{2}(-e_2^1 \cdot e_3^2 - e_2^2 \cdot e_3^1), \tag{3.19}$$

and E_g is symmetric under interchange of indices, as desired.

Note that, since we have made approximations in the definition of the deflection angles, the springs are nonlinear, as was mentioned in section 3.1.1.

3.1.3 Energy Minimization

We now turn to the problem of, given a raw state, how to find the minimum energy configuration of that state, in order to find the corresponding optimized state. For simplicity, we use simple gradient descent optimization with fixed step size. Although this does not guarantee convergence to an optimum solution in all cases, in particular if the starting point is outside the basin of attraction of the optimum, in this case it seems to perform well. This is due to the fact that when a new node is added to the structure, it is initially positioned in its nominal position relative to the node to which it is first connected. This ensures that nodes start "close" to their optimal positions, thus minimizing the likelihood that the gradient descent optimization will diverge from the optimum.

To do gradient descent optimization, we first compute the gradient of the energy function. Since during geometric optimization, the number of nodes and bonds is fixed, we need only the gradient of E_g.

Each bond asserts both a force and a torque on each the two nodes between which it exists. These forces and torques are antisymmetric: $F_{12} = -F_{21}$, and $\mathcal{T}_{12} = -\mathcal{T}_{21}$. We compute the force by taking the gradient of E_g with respect to X, and the torque by taking the gradient of E_g with respect to the three twist angles.

$$E_g(X^1, O^1, X^2, O^2) = \frac{1}{2}k_{sep}l_{sep}^2 + \frac{1}{2}k_{rad}\Theta_{rad}^2 + \frac{1}{2}k_{tang}\Theta_{tang}^2 + \frac{1}{2}k_{twist}\Theta_{twist}^2, \quad (3.20)$$

so

$$DE_g = k_{sep}l_{sep}Dl_{sep} + k_{rad}\Theta_{rad}D\Theta_{rad} + k_{tang}\Theta_{tang}D\Theta_{tang} + k_{twist}\Theta_{twist}D\Theta_{twist}. \quad (3.21)$$

Now, at this point we will make some approximations to streamline the computations involved in doing the optimization: Let us assume that the angle between the two binding sites' local coordinate systems is small. In this case, it is reasonable to define a nominal coordinate system $(\bar{e}_1, \bar{e}_2, \bar{e}_3)$, with respect to which the forces and torques on the binding sites will be expressed, and which has the following properties:

$e_1^1 \cdot \bar{e}_1 \approx +1$	$e_1^2 \cdot \bar{e}_1 \approx -1$
$e_2^1 \cdot \bar{e}_2 \approx +1$	$e_2^2 \cdot \bar{e}_2 \approx +1$
$e_3^1 \cdot \bar{e}_3 \approx +1$	$e_3^2 \cdot \bar{e}_3 \approx -1$
$e_i^1 \cdot \bar{e}_j \approx 0 \qquad i \neq j$	$e_i^2 \cdot \bar{e}_j \approx 0 \qquad i \neq j$

Now, $Dl_{sep} = DX^2 - DX^1 + De_1^2 - De_1^1$. If we write De_1^1 and De_1^2 in terms of differential rotations $D\omega$, about the axes of their respective local coordinate systems,

$$\begin{aligned} De_1^1 &= -D\omega_{e_2^1} e_3^1 + D\omega_{e_3^1} e_2^1 && (3.22)\\ &\approx -D\omega_{\bar{e}_2} \bar{e}_3 + D\omega_{\bar{e}_3} \bar{e}_2 && (3.23)\\ De_1^2 &= -D\omega_{e_2^2} e_3^2 + D\omega_{e_3^2} e_2^2 && (3.24)\\ &\approx D\omega_{\bar{e}_2} \bar{e}_3 - D\omega_{\bar{e}_3} \bar{e}_2, && (3.25) \end{aligned}$$

we find

$$De_1^2 - De_1^1 \approx 2(D\omega_{\bar{e}_2} \bar{e}_3 - D\omega_{\bar{e}_3} \bar{e}_2). \qquad (3.26)$$

And if we further reexpress approximations to the differential twist angles:

$$\begin{aligned} D\omega_{\bar{e}_1} &\approx +D\Theta_{twist} && (3.27)\\ D\omega_{\bar{e}_2} &\approx -D\Theta_{tang} && (3.28)\\ D\omega_{\bar{e}_3} &\approx +D\Theta_{rad}, && (3.29) \end{aligned}$$

then we may write

$$\begin{aligned} DE_g &\approx k_{sep} l_{sep} (DX^2 - DX^1 + 2(D\omega_{\bar{e}_2} \bar{e}_3 - D\omega_{\bar{e}_3} \bar{e}_2)) && (3.30)\\ &+ k_{rad} \Theta_{rad} D\omega_{\bar{e}_3} && (3.31)\\ &- k_{tang} \Theta_{tang} D\omega_{\bar{e}_2} && (3.32)\\ &+ k_{twist} \Theta_{twist} D\omega_{\bar{e}_1}. && (3.33) \end{aligned}$$

Therefore, an approximate gradient for E_g can be written:

$$\begin{aligned} F_{approx} &= -k_{sep} l_{sep} && (3.34)\\ \mathcal{T}_{approx} &= \begin{bmatrix} k_{twist} \Theta_{twist} \\ -k_{tang} \Theta_{tang} + 2k_{sep} \bar{e}_3 \cdot l_{sep} \\ k_{rad} \Theta_{rad} - 2k_{sep} \bar{e}_2 \cdot l_{sep} \end{bmatrix} && (3.35) \end{aligned}$$

where $\mathcal{T}_{approx}$ is expressed component-wise in the basis $\begin{bmatrix} \bar{e}_1 \\ \bar{e}_2 \\ \bar{e}_3 \end{bmatrix}$.

Let me conclude by pointing out once again the caveats of this development:

- The three angular springs are nonlinear, since we dropped the sin() from the bilinear form of 3.7. They are approximately linear, however, for small perturbations from nominal position.
- The three angular springs may interact in general, but again this effect is small for small perturbations.
- Finally, the choice of a "nominal" coordinate system in which to express the gradient *approximately* may seem suspect, since after all, the gradient of the energy is well defined. The key point to realize here is that, as long as the coordinate system in which the downward step is taken is *approximately* equivalent to the coordinate system in which the gradient is naturally expressed, the step taken will still be downhill, and the energy will still converge to the minimum.

As has been noted above, none of these simplifications is a disaster in practice, and the computational simplification thereby afforded is great.

3.1.4 Parameters

The parameters of the TFM are shown in the table below. Of course, these parameters are per binding site, but after the same fashion of [1, 2, 4, 23, 17] , the parameters for each binding site of a new node are copied from a "prototype" node. The prototype nodes with all associated parameters, one for each Berger-Shor node type, are specified by the user.

parameter	meaning
k_{sep}	separation spring constant
k_{twist}	twist spring constant
k_{rad}	radial bending spring constant
k_{tang}	tangential bending spring constant

3.2 The TFM is Constructive

I would like to stress again at this point that this model is *constructive*, in the sense that it suggests an actual assembly pathway for capsids, rather than just demonstrating some principle of optimality. This constructiveness is shared with, and inspired by, the earlier work of Berger *et al.* [1, 2, 4, 23, 17] but attempts to explain things at a more detailed physical level.

Compare this with the work of [15] and [25], which both start with a predetermined number of monomers constrained to the surface of a sphere and optimize the monomer positions relative to that restriction. While they may demonstrate that capsids of the forms found in nature are optimal with respect to some criterion, they make no attempt to explain how such capsids come to be that way.

It is interesting to note that the original work of [5], while stating the principle of optimality which came to be known as quasi-equivalence, also alluded in one of the concluding paragraphs to the idea of a "cumulative strain" model of construction. But to my knowledge, no serious attempt has been made in the meantime to propose a cumulative strain model.

Hopefully, the TFM will go some small way toward bridging the gap between phenomenological models of capsid structure, and the underlying physics of assembly.

Chapter 4

Simulator Design

4.1 Introduction

4.1.1 The Berger-Muir Simulator

The simulator used for this work is based on the simulator of Berger and Muir [17], later modified by Berger and Schwartz [23]. This was a sequential assembly simulator, as originally written. It first read in a file describing the Berger-Shor node types and their associated geometries. It then started with a single "root" node, and conducted a breadth-first search or randomized search of unconnected binding sites.

When an unconnected binding site was reached in the search order (hereafter referred to as the "parent" binding site), a new "child" node of appropriate type (according to the Berger-Shor rules) was created, moved to its nominal position relative to the parent binding site, and connected to the parent. The entire structure was then searched for "nearby" pairs of unconnected binding sites of compatible types, and any pairs so found were connected. The entire structure was then geometrically optimized by gradient descent optimization, and finally the search moved on to the next chosen binding site.

This process continued until no more binding sites were available, or until there was insufficient room at every available binding site to squeeze in more nodes. At this point, the user was given a chance to interact with the (at this point static) structure, by rotating and zooming the view. A simple command line interface was provided, for setting some important simulation parameters.

This simulator was successfully used to demonstrate the feasibility of the Berger-Shor local rules hypothesis. The emphasis at this stage was on implementing the local rules theory, and geometric optimization played a relatively minor role: Since there were as many different node geometries as binding environments, each node could fit more or less perfectly into its binding environment. Optimization was only necessary to "repair" minor cracks in the capsid due to accumulated geometric error from many compounded assembly operations. Later, optimization was used more aggressively, to test the robustness of the local rules against random geometric perturbations [4].

That geometric optimization did not play a central role in simulation at this stage is only reasonable, since strain was not part of the theory of capsid formation being proposed. The combinatorial structure of the local rules was responsible for producing global symmetry, and optimization was added only to facilitate simulation.

4.1.2 The Berger-Schwartz Simulator

Berger and Schwartz later modified the simulator to test a variety of derivative, more detailed, hypotheses; most of which were motivated by comparing results obtained with the previous simulator with laboratory data on natural viruses.

Specifically, the following facilities were added:

- An option to use a randomized search order, rather than breadth first search.
- An option to allow probabilistic breaking of bonds, in order to be able to do more detailed kinetic analyses.

- An option to begin capsid growth in one of a selected few predetermined states, specifically a single hexamer or a single pentamer.
- An option to allow randomized variation of the rule geometries, before simulation began, in order to test robustness of the rules.
- A facility to allow the simulation of the effects of "scaffold proteins".
- A facility to allow dynamic reconfiguration of node types, once they were already added to the structure.

The results with this extended simulator were outstanding, and have contributed greatly to our understanding of the assembly process in nature. Among the most important results:

- It was shown that a spurious hexamer (a hexamer located where a pentamer "should" be) could cause malformations of the nascent capsid similar to those observed in nature; most notably the "spiral" malformation [4]. It has also been shown by Schwartz *et al.* that a node missing one bond can also cause a spiral malformation [24].
- It was also shown that an ambiguous set of Berger-Shor rules, along with one supplementary disambiguating rule of a different type, are sufficient to specify the formation of *both* $T = 4$ and $T = 7$ capsids, depending on the choice of the disambiguating rule. This is interesting, since the $T = 4$ and $T = 7$ capsids are known to be closely related [14, 12]: This "local rule switching mechanism" [2] was proposed as a possible explanation for this relation.
- Further, along the lines of the previous item, facility for "scaffold proteins" was added to the Berger-Muir simulator, and it was shown [4, 2] how the "disambiguating rule" mentioned above might come about physically, through application of scaffold proteins. The proposed locations of the scaffold proteins were later experimentally verified by Thuman-Commike *et al.* [26].

4.2 Redesign of the Simulator

4.2.1 Motivating Factors

The key points of the above discussion, for purposes of this work, include:

- The original simulator placed little emphasis on geometric optimization, because it was not a key feature of the underlying hypotheses being tested.

- The simulator had been modified so many times by this point, to do so many specific tasks, that the simple command line option-based interface had become cumbersome.

- The simulator was not highly interactive: It did not allow pausing and resumption of simulation, much less interactive view changes, loading and saving of state, or fine-grained interactive control over simulation parameters or state.

It was necessary to solve these problems in order to have a simulator versatile enough to develop the TFM in detail. To this end, I undertook the task of rewriting large portions of the Berger-Schwartz simulator, with the following goals in mind:

- The simulator should have built-in support for a more general geometric energy model, at least flexible enough to support detailed experimentation with the TFM.

- The simulator should have an interactive user interface which provides direct control over parameters, and allows direct control of, and interaction with, the simulation in progress. Support should also be provided for saving and restoring simulation state.

4.2.2 Support for TFM

To provide support for the TFM, it was first necessary to extend the file format specification for rule input, to allow specification of the four different spring constants for each type of binding site. This was done by extending the set of option tokens already available, to include an option token for each of the different spring constants.

If any of these tokens are not specified, the spring constant reverts to some reasonable default value. In this way, the new simulator preserves backward compatibility with rules files from the previous simulator.

More importantly, the geometric optimizer had to be generalized to handle all the new parameters. For this, I first rewrote the gradient descent optimization code to use the new gradient with all the TFM parameters included, as described in chapter 3.

This worked for short simulation runs, but during more extended tests a problem related to optimization appeared, which I will now describe.

The gradient descent optimizer moves and reorients each node by a small increment, at each optimization step. Since the orientation of a node is implicit in the orientations of its binding sites, the reorientation consisted of rotating the binding sites themselves.

Now, recall from chapter 3, that the orientation of a binding site is represented by a matrix whose column vectors represent the e_1, e_2, and e_3 vectors of the binding site. Rotating such a site involves computing a rotation matrix and multiplying it by the orientation matrix, resulting in a new orientation matrix. Since the original orientation matrix is orthonormal, and so is the rotation matrix, the new orientation matrix should also be orthonormal.

However, a typical simulation run may require rotating such a matrix several tens of thousands of times. In this case, round off error can and does accumulate, causing the orientation matrices to drift far from orthonormality. This eventually causes the

nonlinear springs to go far outside their linear range, and at this point the entire simulation can go quite irreparably awry.

To circumvent this problem, I periodically pause during optimization and re-orthogonalize the orientation matrices. The reorthogonalization method is borrowed from the computer graphics community, and is briefly described in [22] and [20]. I summarize as follows:

To reorthogonalize a matrix R, compute an approximation to a "correction matrix" $C = (R^T R)^{-\frac{1}{2}}$. The product RC is then orthogonal. To compute an approximation to C, we use the fact that $R^T R$ is close to I, and use a Taylor expansion to find:

$$C \approx I - \frac{1}{2}X + \frac{3}{8}X^2 + ... \tag{4.1}$$

The simulator currently performs this reorthogonalization on each binding site every 100 optimization steps. This frequency is *ad hoc*, but effectively eliminates the orthonormality drift problem and does not add more than one percent of computational overhead.

The question I have not addressed is that of drift of the binding sites *relative to each other*. So far, this has not been a problem, but if I were to attack it, my approach would be to change the representation of a binding site to be expressed in a "parent" coordinate system of the node on which the binding site occurs. Then, instead of having to rotate all binding sites in the structure during a gradient descent step, we could simply rotate the parent coordinate system of each node.

However, since this would require a change of data representation and hence (since the simulator was not originally written in object oriented style) a complete rewrite of all simulator code which relates to geometry, I deemed it too much work for little benefit; especially since the simulator now performs well, even on long runs.

4.2.3 User Interface / Capabilities

The user interface to this simulator is completely new, and is significantly more powerful than the old:

Figure 4.1: The simulator control panel, in its current configuration.

Window-Based The interface is written in TCL/TK [19], and is window-based to provide a fast learning curve for new users, and to allow direct viewing and interactive setting of important simulation parameters. Such parameters include the simulation temperature, the various geometric thresholds for connecting nodes, the gradient descent step size, the bond energy, the Gibbs energy, etc. A picture of the simulator control panel, as currently configured, is shown in figure 4.1.

Fine-Grained Simulation Control The new user interface also provides direct control over the progress of the simulation, also via the control panel. In particular, facility is made for pausing, resuming, and single stepping the simulation, as well as for reinitialization. Facility is also made for control of various simulator modes, such as disabling repeated display during geometric optimization (for faster simulation runs), disabling the display entirely (for even faster runs), *etc.*

View Control The new UI also allows the user to select an arbitrary viewing direction, when the simulation is paused, simply by clicking and dragging the structure. This, combined with the pause and single step features, facilitates better inspection

and understanding of the detailed dynamics of structure formation. With the old simulator, it was often difficult or impossible to see the detailed causes of many problems during assembly, simply because the simulation was proceeding too fast, or because the particular location was visually obscured by other parts of the structure. These new features provide an effective solution to this problem.

Direct Structural Interaction An extremely important extension for experimentation, is the new user interface's feature set for direct, visually directed interaction with the structure: A mode is available wherein the user, by pointing and clicking on nodes and binding sites in the graphical representation of the structure, may *select* such nodes and sites for further operations. Examples of some operations which have been built using this feature, and which are currently available, are adding and deleting nodes at specific locations; making and breaking specific bonds; and node and binding site queries, in which important textual information pertaining to the selected node or binding site is dumped to the output stream. It should be readily apparent how powerful these features are for investigating different models and hypotheses about capsid assembly.

State Saving and Restoration In case the user wishes to pause and later resume work, facility has been made for saving the current topology to disk, and for later retrieval and reconstruction of the same topology. This feature can also be used for saving an important intermediate topology, in order to return to it later and try a different experiment thenceforth.

This facility may also be used in the following manner: The user may specify one set of node geometries, construct a structure using them, and save the topology. Then, she may wish to construct a *different* set of node geometries, and construct the same topology using this different set, by loading the topology from disk. In this case, the topology may not be a possible result of the second set of geometries, so

this gives one a method of investigating "unreachable" topologies for a given set of rules.

The file format for topology dumps is text-based, and easy to understand. So another possible application is construction of topologies to specification, using a text editor, after which they may be loaded into the simulator using a given set of rules, and investigated in this context.

Command Line Interpreter In addition to the window-based (TK) user interface, the simulator also provides a text-based, fully programmable, TCL command line interpreter. This affords the skilled user greater flexibility and precision in controlling the simulator, as well as providing extensibility and configurability: The behavior of the simulator can be changed, and new commands can be added. In fact, the existing control panel is constructed by a script, interpreted by the command-line interpreter at startup, so altering or extending the control panel is a simple and quick task.

4.3 The Problem of Metastable States

The TFM simulator was based on an intermediate version of the Berger-Schwartz simulator, which did not yet have extensive capabilities for backtracking [1]. As a result of this, as experiments with the simulator (modified as described in sections 4.2.2 and 4.2.3) progressed, it became apparent that the simple sequential (*i.e.*, non-backtracking) construction algorithm was severely limiting the quality of results.

Most notably, since the TFM uses only one type of Berger-Shor node, there exist vastly many topologies which satisfy the binding rules (since the binding rules are, in this case, almost trivial). The vast majority of these are metastable states. A metastable state is a state which was arrived at by a legal series of simulation moves,

[1] Please note that this simulator has since been improved by Schwartz *et al.*, to better facilitate backtracking during the search procedure.

with geometric optimization after each move, but whose energy is nevertheless not minimal for that number of simulator moves (or nodes).

Clearly, in a simulator with no means for backtracking, and with an energy model in which metastable states exist, there is no guarantee that the global energy minimum can ever be reached.

Moreover, a sequential simulation is not a good model of physical reality, since in solvent-based chemical systems, thermal kinetic energy can excite the system and allow it to escape from metastable states and approach the Maxwell-Boltzmann distribution, which is its steady-state probability distribution.

Accordingly, we would like to modify the simulator to produce results which are distributed according to the Maxwell-Boltzmann distribution. If we could do this, then we would know whether the TFM, as proposed, produces a steady-state probability distribution which favors the symmetry observed in nature. If so, then it seems reasonable to assert that the TFM is one possible explanation for the observed natural symmetry of capsids.

Fortunately, there is a reasonably easy way to modify the simulator to do this, and that is to use the Metropolis Method. This is the subject of the next chapter.

Chapter 5

The TFM and the Metropolis Method

In this chapter, I first introduce the Markov Chain Monte Carlo method in general, as a sampling method. I then move on to a specific application of MCMC, the Metropolis Method, which can be used when the distribution to be sampled from is a Maxwell-Boltzmann distribution. Then I discuss the specifics of application of the Metropolis Method to the TFM, both in terms of simulator implementation and, more importantly, how and why we may adopt various interpretations of the simulation results.

5.1 Introduction to the Markov Chain Monte Carlo Method

The Markov Chain Monte Carlo (or MCMC) method is a general paradigm for the design of approximation algorithms which attempt to sample from some very large

state space, Ω, according to some probability distribution, $\pi(\cdot)$. Some example areas in which such problems occur are:

- *Statistical physics:* Here, we wish to find expected values of observable state functions. That is, we have some function F, defined on Ω, and we wish to find the expected value of F with respect to the probability distribution π.

- *Combinatorial Optimization:* Here, we wish to find an approximate solution to a combinatorial optimization problem. That is, Ω is the space of feasible solutions, and we have again a function F, defined on Ω, and whose value we wish to optimize. In this case, we may construct π in such a way that "better" feasible solutions to the optimization are favored. Thus, sampling from π gives us an approximate solution to the optimization problem.

The problem of discovering which structures are formed by the TFM can be considered as either of the above types: We may wish to ask the question, "What is the expected value of a particular indicator function f, defined on Ω." Such a function f could, for instance, be one for perfect capsids, zero otherwise, or it could be one for structures with the "correct" number of monomers, zero otherwise, *etc.* These are all instances of finding the expected value of an observable; this is the first problem above.

Or, we may wish to consider the TFM problem as finding the answer to the question of what is the lowest energy state reachable by building structures using monomers of a certain geometry. This is a problem of the second type mentioned above, a combinatorial optimization problem.

A more complete and detailed description of MCMC is given in [10] and elsewhere, but I will give a brief summary here: The idea of MCMC is to construct a Markov Chain, $\mathcal{M}$, with state space Ω, such that:

- $\mathcal{M}$ is ergodic and has stationary distribution π. In the case of Markov Chains, to say that $\mathcal{M}$ is ergodic is equivalent to saying that it is aperiodic and connected.

- The transitions of $\mathcal{M}$ are simple perturbations of structures corresponding to the states in Ω, and hence the next state is simple to compute based on the previous state.

Then, if we wish to average a state function, we simply use an estimator for the expectation of the function F:

$$\langle F \rangle \approx \frac{1}{N} \sum_{i=1}^{N} F(\omega_i), \tag{5.1}$$

where the ω_i's are nearly independent samples from π. On the other hand, if we wish to do optimization, it suffices to take the best value of $F(\omega_i)$, also for some number of nearly independent samples, $\omega_i, \quad i = 1, ..., N$.

In either case, we can obtain nearly independent samples by running $\mathcal{M}$ for some number of time steps, T, and returning the resulting state. The hope is that if T is large enough, the probability distribution of the resulting final state will be close to π, the equilibrium distribution.

5.1.1 Setting up the Markov Chain

Let us define the Markov Chain $\mathcal{M}$, and then we will show that it meets the requirements.

At time step t, we are in some state, call it ω_t. We wish to define the transition probabilities

$$A(x, y) \equiv P(\omega_{t+1} = y | \omega_t = x), \tag{5.2}$$

for all x and y in Ω. This is the conditional transition probability from state x to state y.

To define $A(\cdot,\cdot)$, let us first pick a "candidate state" ω_{cand}, from some candidate distribution Q which may be dependent on the current state:

$$\omega_{cand} \sim Q(\,\cdot\,;\omega_t). \tag{5.3}$$

Now, evaluate $\pi(\omega_{cand})$ and $\pi(\omega_t)$, and let:

$$\omega_{t+1} = \begin{cases} \omega_{cand} & \text{with probability} \quad \min\left(1, \frac{\pi(\omega_{cand})Q(\omega_t;\omega_{cand})}{\pi(\omega_t)Q(\omega_{cand};\omega_t)}\right) \\ \omega_t & \text{otherwise} \end{cases} \tag{5.4}$$

The number, $\min\left(1, \frac{\pi(\omega_{cand})Q(\omega_t;\omega_{cand})}{\pi(\omega_t)Q(\omega_{cand};\omega_t)}\right)$, is typically called the *acceptance probability.*

With a bit of algebra (see for instance [18]), it can be shown that this method of choosing the successor state, ω_{t+1}, given ω_t, satisfies the condition of *detailed balance*:

$$\pi(x)A(x,y) = \pi(y)A(y,x) \quad \forall x, y. \tag{5.5}$$

A Markov Chain $\mathcal{M}$ satisfying this condition is said to be *reversible.* If we choose the candidate distribution, Q such that every state is reachable from any other state, then $\mathcal{M}$ is further said to be *connected.* Reversibility and connectedness together imply ergodicity, hence if we choose Q and A as described, $\mathcal{M}$ will be ergodic and therefore converges to its (unique) stationary probability distribution, $\pi(\cdot)$.

This is good, since now if we run $\mathcal{M}$ for a long enough while and choose then the final state, we have a sample from a probability distribution which is very close to $\pi(\cdot)$. By choosing many such samples, we may integrate with respect to $\pi(\cdot)$, sample from $\pi(\cdot)$, or optimize a function on Ω, as described above.

5.2 The Metropolis Method

The Metropolis Method, first described in [16], is a classical example of the Markov Chain Monte Carlo method, applied to the problem of finding the values of state functions of an equilibrium thermodynamic system.

An equilibrium thermodynamic system has a microstate S, distributed according to the Maxwell-Boltzmann distribution:

$$\pi(S) = \frac{1}{Z} e^{-\frac{E(S)}{kT}}, \tag{5.6}$$

where k is Boltzmann's constant, T is the temperature, $E(\cdot)$ is the energy function defined on states, and Z is the normalizing factor which makes P a probability distribution:

$$Z \equiv \sum_{S \in \Omega} e^{-\frac{E(S)}{kT}}. \tag{5.7}$$

Z is called, in statistical physics parlance, the *partition function*. Evaluating Z exactly is, in general, computationally intractable, since Ω can be extremely large.

If Z is unknown, one cannot evaluate $\pi(\cdot)$. This appears at first to be a problem. However, due to the form of equation 5.4, if π is a Maxwell-Boltzmann distribution, the partition function cancels when we form the ratio $\frac{\pi(\omega_{cand})}{\pi(\omega_t)}$.

Hence in this case, we need not know Z to compute the acceptance probability. The acceptance probability is simply

$$\min(1, \frac{Q(\omega_t; \omega_{cand})}{Q(\omega_{cand}; \omega_t)} e^{-\frac{\Delta E}{kT}}), \tag{5.8}$$

where ΔE denotes the energy difference $E(\omega_{cand}) - E(\omega_t$.

In a special case of the Metropolis Method, if $Q(\cdot; \omega)$ is uniform (and hence also independent of ω), the acceptance probability reduces to

$$\min(1, e^{-\frac{\Delta E}{kT}}). \tag{5.9}$$

This is the form used, for example, in the Boltzmann Machine [18].

5.3 Application of the Metropolis Method to the TFM

I stated earlier in section 4.3, that I applied the Metropolis Method to the TFM simulator in order to alleviate the problem of getting stuck in "Metastable States". This alone would be justification enough for applying Metropolis, if we were only concerned with optimization. But many researchers (see chapter 1) have already proposed optimization-driven models of capsid assembly. By contrast, we seek here an operational model, one which more explicitly models the physical processes through which capsids form.

It therefore behooves us to consider exactly *in what way* we are modeling physics when we run this Markov Chain, so that we may interpret the results of the modeling in an appropriate context.

5.3.1 Interpretation

Samples from Equilibrium Distribution: One simple way to interpret the Metropolis simulation is to run multiple simulations, each for a long time, and interpret the results of each simulation run as an independent sample from the equilibrium probability distribution of a single capsid.

In other words, if the TFM were to be a "correct" model, and we were to observe a single formed capsid, we would expect the state of such a capsid, interpreted as a random variable, to be distributed according to the distribution given by the outcomes of many independent simulation runs. Thus, use of the Metropolis Method in this way gives us a way of evaluating the likelihood of any particular outcome (such as perfect capsids, for example), assuming the TFM is correct.

Ensemble Samples: We can also interpret the results of independent simulation runs as independent samples from an ensemble of capsids. In other words, if we look at a large population of capsids, the fraction of that population which we expect to observe in some class of states, $\mathcal{C} \subset \Omega$, should be the same as the probability of a single simulation run resulting in a state from the class $\mathcal{C}$.

This gives us another way of interpreting the simulation results. Note, however, that since $\mathcal{M}$ is ergodic, these two interpretations are identical, as long as we are careful about running the simulation long enough to produce an independent or nearly independent sample from π each time.

Construction Sequence: A third and perhaps more valuable way to interpret the Metropolis simulation is as a *dynamical* simulation, with the progress of the simulation representing actual time progress of the formation of a capsid. This interpretation is allowed by the fact that the "moves" used in the simulation reflect incremental chemical changes to the structure (see section 5.3.2): We can think of the sequence of moves made by the simulator as representing a series of chemical events which occur sequentially to produce the final structure.

Because the moves are chosen to correspond to chemical events, only a small portion of Ω is reachable from any given state ω, in one time step. In particular, the candidate distribution Q is nonzero for only those states which differ from ω by one "operation". This is in general a *very* small subset of Ω. Contrast this, for instance, with the method of Gibbs Sampling [18], in which the candidate distribution is supported by all of Ω: In one time step, any state can be reached from any other state. These two methods will both converge to the same stationary distribution; the difference is in *how fast* they will do so. The method of using only moves corresponding to chemical events will in general take much longer to traverse Ω.

Why then should we use this slower method? The answer is that it is physically more reasonable: The time required for the simulation to reach a given state is more

closely related to the time required for a physical system to reach the same state, since the method of exploration of Ω (the candidate distribution Q), is the same. The simulation time and the actual time are not exactly proportional, however, since in a physical system multiple events may occur simultaneously at different physical locations, whereas the simulation is inherently sequential.

If we wished only to optimize the energy with respect to state, we would use the fastest optimization technique available. However, the optimum state so discovered might be one which is separated from the initial state of the system by an energy barrier so high or so wide as to be effectively unreachable by the physical system being modeled. Consider the case of a spin glass at a temperature T_0, at which it is effectively quenched. It will effectively never reach its optimum energy state as long as it stays at that temperature, and a Metropolis simulation using single spin flips will also remain so quenched. However, if we were to use simulated annealing, we could maybe find the optimum state within a reasonable amount of time. But this would be "cheating" in a sense which we wish to avoid: Since we are trying to develop an operational model, we should stick with a simulation technique which is physically reasonable. This holds not only for the energy function, but for the simulation dynamics as well, and that is why I have chosen the moves in the Metropolis simulation to be so restricted.

5.3.2 Implementation

Let us now turn to the details of the implementation of the Metropolis Method used in the TFM simulator.

Candidate Distribution

We first address the question of what the moves actually are, which I am claiming represent chemical events.

At each time step, the state ω_t of the simulation is given by a collection of nodes with indexed binding sites, connected in a graph structure. Additionally, the nodes have positions and orientations in $\mathcal{R}^3$, but this will not directly concern us here.

The procedure for generating a sample from Q, the candidate distribution, works as follows:

```
procedure pick_candidate_state {
  pick a node N, uniformly at random
  pick a binding site B on N, also uniformly
  if B is unconnected {
    if another binding site B2 is in range and has the correct type {
      connect B to B2
    } else {
      make a new node N2,
      orient it appropriately, and connect it to B
      if any more nodes are in connection range of N2 {
        connect them also to N2
      }
    }
  } else {  /* B is connected */
    delete N from the structure
```

```
    if the structure is now disconnected {
      pick one fragment to keep
      delete the rest
    }
  }
}
```

This results in a candidate state which usually differs from ω_t by only a small number of nodes and bonds, with the exception of the case when the structure ends up fragmented after a node is removed. In this case, a single fragment is chosen to survive and the others are discarded. This was necessary to ensure a simpler implementation: The simulator need only deal with one connected set of nodes at any time. Contrast this with the simulator of [23], for instance, which is able to deal with arbitrarily complex "soups" of freely floating structures (though it also has an option for restricting assembly to a single structure).

It would appear that the discarding of disconnected fragments would render the simulation substantially less realistic. But a moment of thought, and experience with the simulator, shows that this is not actually so: As the structure begins to grow initially, it has a spiny, treelike structure, and its connectivity is small. In this initial phase, fragments can be and often are discarded; but as the structure grows and its connectivity increases, it becomes more unlikely that removing one node will lead to a disconnected structure, so the probability of having to discard fragments decreases significantly. So the "single fragment" restriction only causes small fragments to be discarded, and these at an early stage of the simulation; it does not seem to be a significant hindrance to the realism of the simulation at advanced stages.

Markov Chain Implementation

We now have our method of choosing a candidate state, ω_{cand}, from an initial state ω_t, as described above. Furthermore, we have our method of evaluating the energy of a state, as described in chapter 4. The application of the Metropolis Method is then a simple matter. We proceed as follows:

```
procedure metropolis_move {
  old_energy = optimize_energy(current_state)
  change current state to candidate state, remembering changes
  new_energy = optimize_energy(candidate_state)
  if (new_energy < old_energy) {
    flush changes   /* candidate is accepted */
  } else {
    accept_prob = exp( - (new_energy - old_energy) / (k * T))
    x = random[0..1]
    if (x < accept_prob) {
      flush changes   /* candidate is accepted */
    } else {
      undo changes /* restore previous state */
    }
  }
}
```

Remembering what changes were made to the current state to arrive at the candidate state may seem a strange approach at first, until one realizes the computational

overhead involved in copying states. If we were to use two states, explicitly stored, we would have to copy an entire state on each time step. But with this method, we need only store enough information to undo the changes which were made to arrive at the candidate state. Since the changes are generally small (see section 5.3.2), this results in far less overhead.

Also, note that the acceptance probability is not computed exactly as is given in equation 5.4, since the ratio of the candidate probabilities,

$$\frac{Q(\omega_t;\omega_{cand})}{Q(\omega_{cand};\omega_t)} \tag{5.10}$$

has been omitted. As above, this may affect realism in the early stages of the simulation, but not in more advanced stages. This is because for two states ω and ω' with large and approximately equal numbers of nodes, $Q(\omega;\omega') \approx Q(\omega';\omega)$, so the ratio is approximately unity and the equilibrium probabilities are not significantly affected.

Chapter 6

Results and Conclusions

The TFM has proven to be a remarkably rich and expressive model, forming many different types of structures depending on the values of the model parameters. In this chapter, I present an overview of the structure of the parameter space, and try to explain qualitatively why the different types of structures are formed with different settings. Following this, I will present some biological hypotheses which seem plausible, but for which I have as yet no evidence. I then make some concluding remarks about this work, and suggest further avenues for research in this area.

6.1 Structures Formed by the TFM

We begin with a (goal-oriented) discussion of some restrictions which must be met by the model parameters, if we are to even hope to produce complete, symmetrical capsids. Certainly, these restrictions will not be sufficient in themselves to guarantee successful capsid assembly, but they are nonetheless necessary.

Some part of these restrictions I predicted ahead of time by inductive reasoning. On the other hand, some have been noted, or refined, by direct experimental observation using the simulator. But I have deliberately chosen to discuss the restrictions

in hindsight, with a synoptic viewpoint, so the reader may better understand their structure.

6.1.1 Critical Restrictions on Parameters

Binding Site Geometry

One of the first facts that becomes apparent when doing experiments with the simulator is that the results obtained are critically dependent on the simulation parameters (the binding site geometries and the flexibility parameters). This criticality manifests itself in the fact that large volumes of parameter space produce malformed structures which have effectively no chance of growing into capsids. Among the problems encountered are:

- **The Too Many Pentamers Problem:** One can choose the angle between the "IN" and "OUT" binding sites in such a way that pentamers are encouraged to form immediately or almost immediately (by choosing the angle between them to be around 108 degrees). However, one can then choose the geometry of the third ("ODD") binding site in such a way that the implicit spherical curvature of the monomer is very small, by choosing the two remaining angles (that between "IN" and "ODD", and that between "OUT" and "ODD") such that their sum is almost $360 - 108 = 252$ degrees. Since there must be exactly twelve pentamers in a capsid restricted to pentamers and hexamers, there is an implicit relation between the pentamer density and capsid radius. In this case, the pentamer density is too high for the curvature of the growing shell, and the structure enters a state of "frustration," in which the edge of the lattice can no longer form closed N-mers (figure 6.1).

- **The Too Few Pentamers Problem:** On the other hand, if the "IN" – "OUT" angle is chosen too large (close to 120 degrees), not enough pentamers will form during the first stages of capsid growth, and a structure will result which contains too few pentamers for the radius of capsid which it defines. In this case also, a frustrated condition typically occurs around the perimeter of the nascent capsid, resulting in stalled growth or in chaotic looking lattices with many unconnected binding sites. This situation is shown in figure 6.2.

So we see that there is an implicit relation between the effective monomer curvature and the IN – OUT angle. This gives us a restriction on the set of feasible monomers from which capsids can be successfully made.

There are other restrictions also. For instance, there is a relation between the implicit "surface area" of a monomer, suitably defined, and its implicit curvature. To see this, note that the surface area of the entire capsid is implicit in the curvature of the monomers which compose it, and that the approximate number of monomers must be the ratio of the capsid surface area to the monomer surface area. But, we have already seen in chapter 2 that the number of monomers in a quasi-equivalent capsid is restricted to a certain class of integers, namely the "T"-numbered multiples of 60. We would therefore expect that capsids can only form from monomers for which the ratio of the implicit capsid surface area to the monomer surface area is approximately one of these numbers. This is borne out by experiment; figure 6.3 shows examples of structures built from two different monomers which differ only in this respect.

Above, we made an observation about the relation of the IN – OUT angle to the sum of the other two angles. We may also ask: What about the relation *between* the other two angles (that is, the IN – ODD angle and the OUT – ODD angle)? The answer is, that holding their sum fixed and varying only their ratio results in a set of structures of constant spherical curvature, but with varying amounts of skew imparted

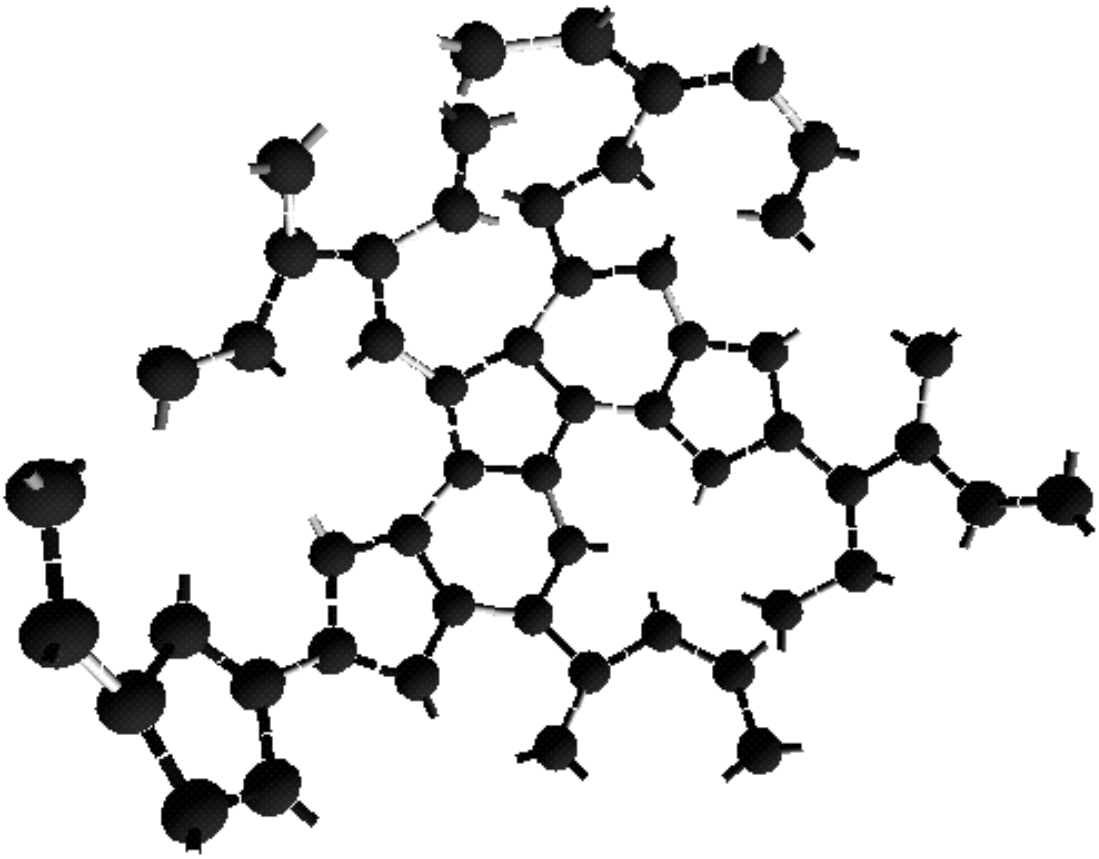

Figure 6.1: When the pentamer density is too high, the nascent capsid encounters a state of "edge frustration".

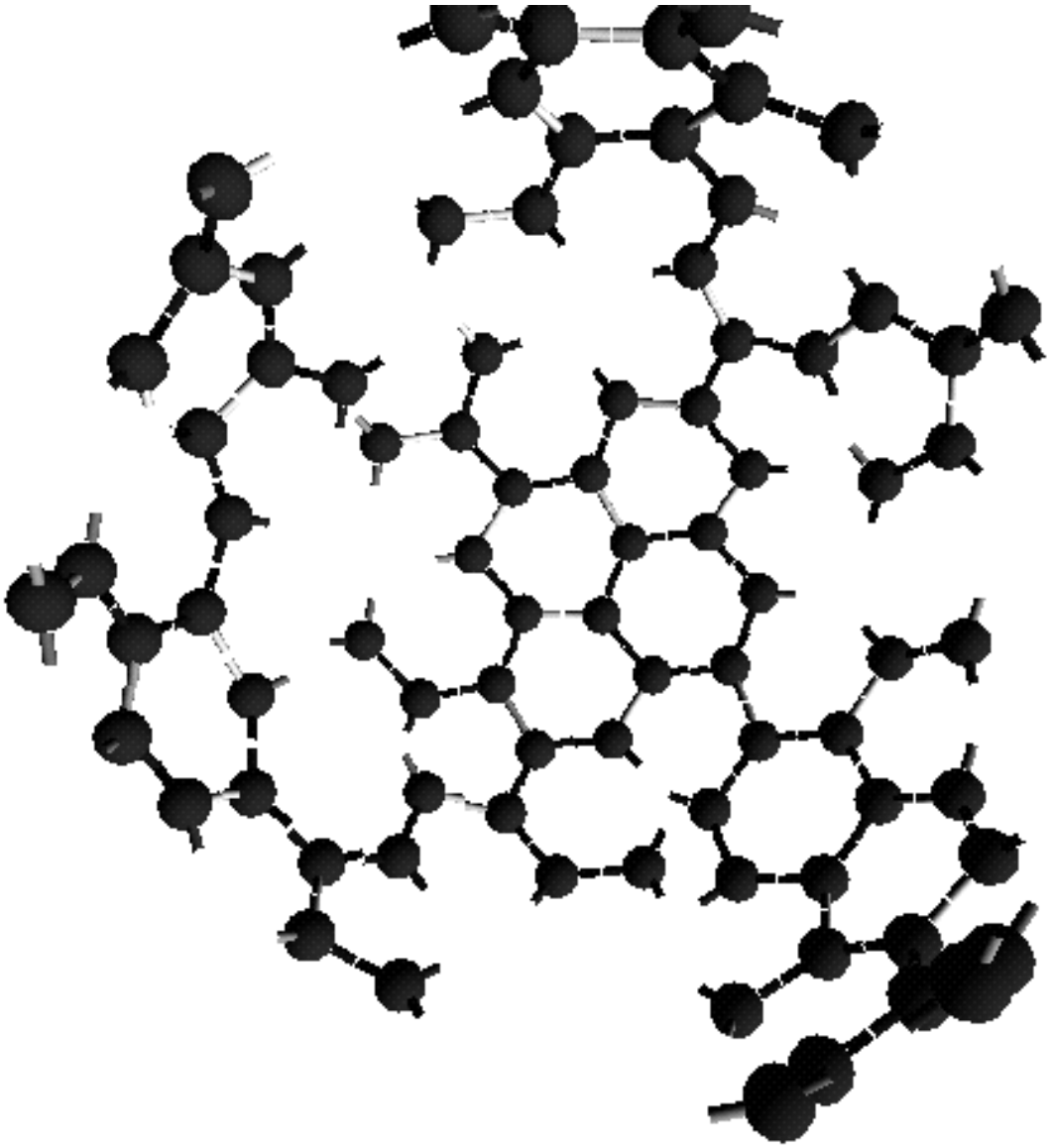

Figure 6.2: When the pentamer density is too low, the nascent capsid typically encounters edge frustration.

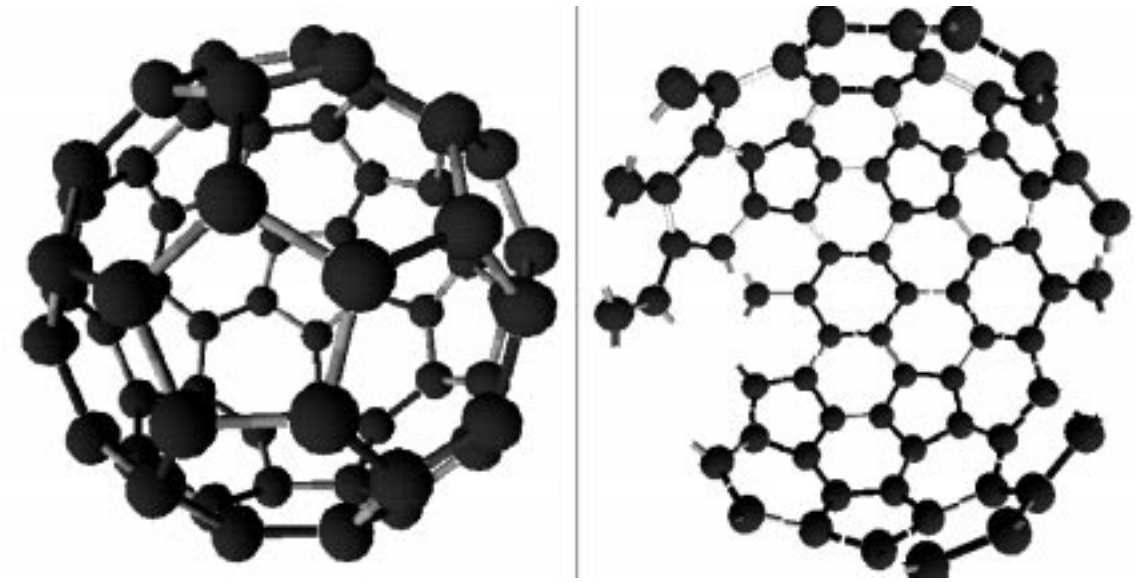

Figure 6.3: The ratio of implicit capsid surface area to monomer surface area must coincide with a valid T-number, if closed capsids are to form. On the left is a capsid which meets this restriction. On the right is an (attempted) capsid for which the implicit capsid surface area has been increased by approximately a factor of two. This capsid never completes.

to the mixed bond hexamers (see figure 6.4). Such hexameric skew is observed in natural capsids (see, for instance, [26]). Its origin and purpose is not well understood, but is hypothesized by Berger and Shor [2] to assist in symmetrical placement of pentamers.

Experience with the simulator leads me to believe that this is in fact the case. To illustrate the role of hexameric skew in pentamer placement, consider a sub-lattice of a hex-lattice as shown in figure 6.4. This is an incomplete $T = 7$ face. Let us assume that the strain has become sufficient at this point that pentamers are about to be introduced. We may ask, at which points of the set { **A, B, C, D, E, F** } will these pentamers form, and by what mechanism will they be symmetrically disposed?

To answer this, let us consider first the case shown on the left. Here, the mixed bond hexamers are unskewed, so all points **A** through **F** are equidistant from **O**; hence, there is equal strain at points **A** through **F**, and no energetic difference produced by introduction of pentamers at any particular locations among these.

On the right, however, the situation is different: Introduction of severe hexameric skew has put points **A, C, E** closer to **O**. Hence if hexamers form initially at **A, C, E**, then the strain at **B, D, F** will be much higher due to their greater radius. Conversely, if pentamers form initially at **A, C, E**, then they will be less likely to form at **B, D, F**, due to strain reduction from pentamers **A, C, E**. So we see that hexameric skew can be instrumental in providing a distinction between topologically equivalent sets of possible pentameric sites, thereby providing an additional restricting factor guiding capsid assembly.

In fact, since the monomer is chiral, the asymmetry between the IN – ODD and OUT – ODD angles is directionally distinguishable. Hence, reversal of the asymmetry between these two angles might produce a capsid enantiomorphic to the original (though I have not done such experiments yet). So hexameric skew provides us also with a possible explanation for the selective chirality of capsids in nature.

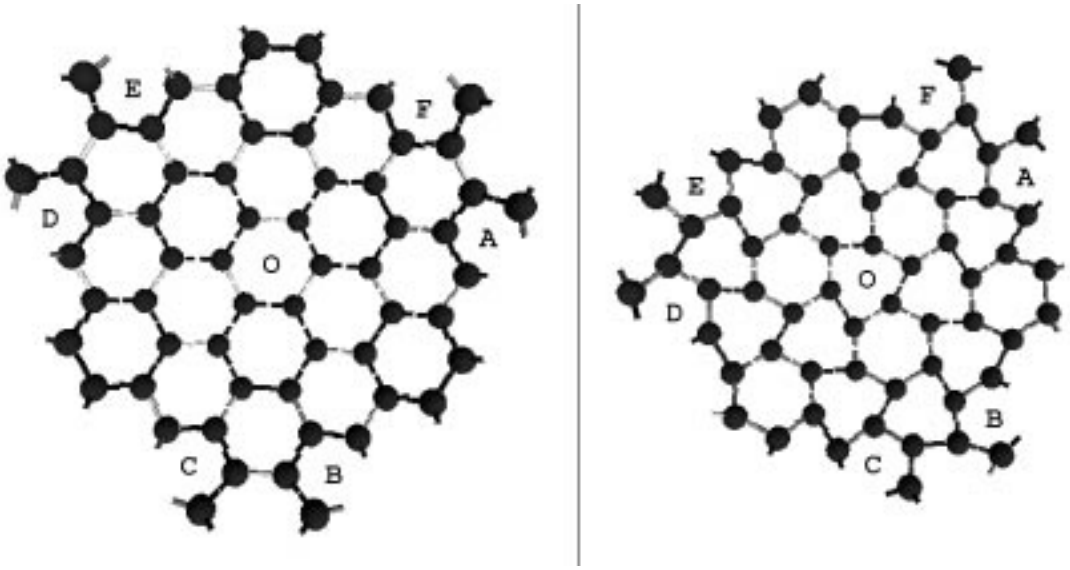

Figure 6.4: An incomplete face of a T=7 capsid, without (left) and with (right) hexameric skew.

Figure 6.5: Too little tangential flexibility can cause the nascent hex lattice to "roll up" into a cylindrical shape. On the left is such a cylinder, viewed perpendicular to its axis. In the middle, the same cylinder viewed parallel to the axis. On the right, an oblique view of the same cylinder.

Flexibility Parameters

From other experiments with the simulator, it is apparent that selective flexibility in the tangential direction, the idea which inspired this work, is in fact necessary for reliable capsid formation. Otherwise, structures are sometimes formed which curve into cylindrical or rumpled shapes, instead of remaining constrained to an approximately spherical surface. See figure 6.5 for an example of such a roughly cylindrical surface. The ratio of radial to tangential flexibilities does not seem to be very critical; it is only important that there is substantially more flexibility in the tangential direction.

On the other hand, in some cases the ratio of tangential flexibilities of the different binding arms does seem to be important. Since pentamer/hexamer switching involves

bending specifically on the "IN" and "OUT"-type binding arms, more tangential flexibility seems to be important on these arms than on the "ODD" arm.

6.1.2 Types of Structures Formed

With these restrictions in hand, I will now discuss the types of structures I have observed during experimentation with the simulator.

Perfect Capsids

One of the most important results, of those so far obtained with the TFM simulator, is that perfectly structured capsids of T-number greater than one can be formed from a single type of monomer. Figure 6.6 shows a perfectly structured $T = 3$ capsid so formed. No higher order capsids have been formed to date, but no extensive attempts have been made. The important thing to realize here is that, if we hold Occam's Razor dear, this experimental evidence speaks strongly in favor of cumulative strain models of capsid assembly: It is hard to imagine a simpler model capable of giving rise to such global symmetry purely on the basis of local interactions. Certainly, producing perfect capsids *only* on the basis of cumulative strain becomes more difficult as the T-number increases, but it would seem that cumulative strain might still be an important one of a number of methods by which high-order capsids self-assemble.

Semi-Perfect Capsids

In addition to perfect capsids, the TFM simulator also produces *semi-perfect* capsids. By this, I mean capsids with a number of monomers of the form $60T$, with all binding sites occupied, and consisting of all pentamers and hexamers (hence also with the correct number of pentamers); but which are perhaps asymmetrical in structure, having one or more misplaced pentamers. An example of such a capsid is shown in figure 6.7.

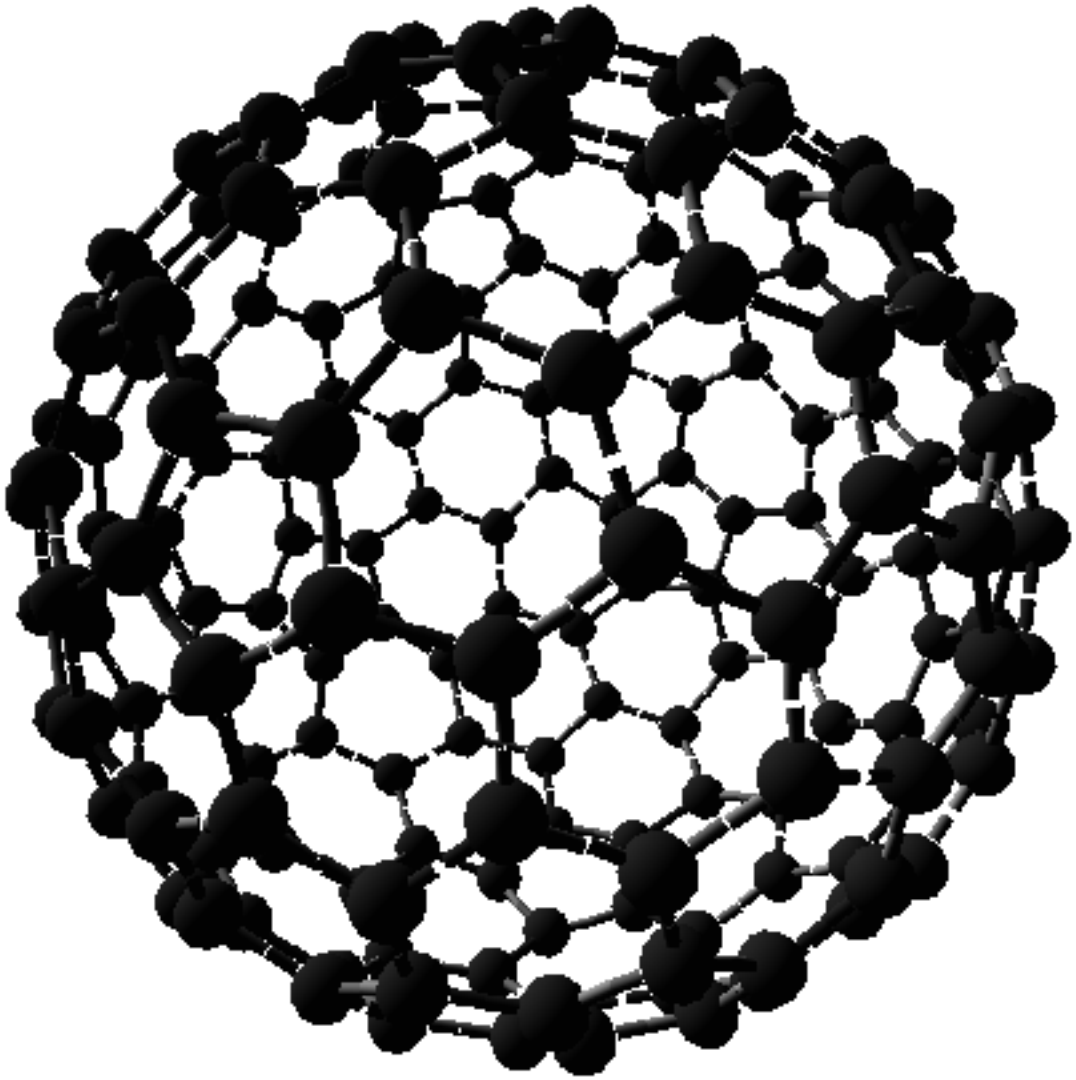

Figure 6.6: A perfectly symmetrical $T = 3$ capsid, produced by the TFM simulator.

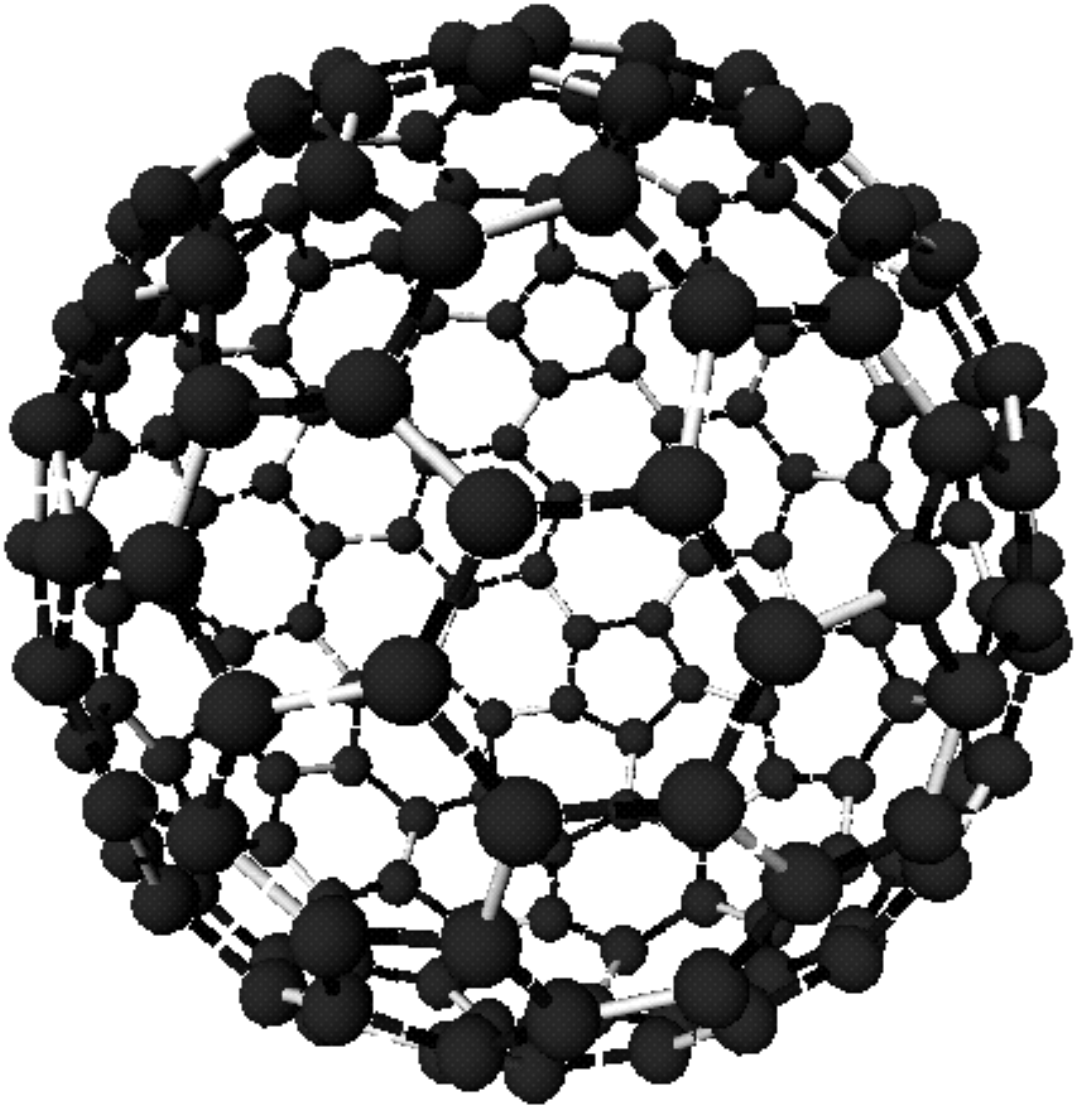

Figure 6.7: A semi-perfect $T = 3$ capsids produced by the TFM simulator. It has $60T = 180$ monomers, all binding sites connected, and consists of only hexamers and (twelve) pentamers, but it is slightly asymmetrical due to misplacement of some pentamers from their ideal positions. Many such capsids have been formed.

These are particularly interesting capsids because they also are observed in nature, perhaps with higher frequency than is implied by a cursory, non-critical reading of the literature [21]. Capsid reconstructions from electron microscopy are typically perfect capsids as presented, but this perfection reflects omission of imperfect samples and substantial averaging, used in the reconstruction process.

Typical reconstruction techniques for quasi-equivalent capsids *assume* icosahedral symmetry and base the results on that assumption. The usual procedure is to obtain 2D projections of lots of capsids, reject the ones which appear malformed (sometimes more than thirty percent!), and then perform a cylindrical Fourier transform on each remaining image. Based on each set of transform data, one may decide the orientation of the capsid in question, again assuming symmetry. One may then reorient each capsid to a fixed, "nominal" orientation, and then average the electron densities to arrive at a complete reconstruction.

It should be obvious that this procedure in its most elementary form does nothing to address the proportion of semi-perfect capsids in nature, nor does it address the types of asymmetries most frequently observed. Moreover, it is unknown whether slightly malformed (*e.g.* perhaps semi-perfect) capsids can form viable, infectious virus particles *in vivo*. It behooves us, then, in the study of capsid structure, to more carefully and completely consider semi-perfect capsids.

Odd Size Capsids

The TFM simulator also tends to produce what I will call *odd size capsids*. An odd size capsid is a closed structure with all binding sites occupied, consisting only of pentamers and hexamers, but which does not consist of a number of monomers equal to $60T$. Usually, the number of monomers is close to $60T$, namely of the form $60T + 6k$, where k is an integer typically between -2 and 2. These, of course, do not have perfect icosahedral symmetry, but are in fact quite similar to the semi-perfect capsids mentioned above, and my comments above apply also to these odd size capsids.

Malformed Structures

As implied above, the TFM simulated in regimes of parameter space not conducive to capsid formation can produce a variety of malformed structures having heptamers, octamers, incorrect pentamer density, edge frustration, saddle-shape, *etc.* In particular, some simulations that produce edge frustration result in structures reminiscent of the "spiral malformation", observed in nature [6, 13, 7]. Such structures, one of which is shown in figure 6.8, are formed when two "leaves" of the nascent capsid grow large separately and then collide, but the structure of the available binding sites proves incompatible. This structure is only *reminiscent* of a spiral malformation because the two mutually exclusive growing leaf structures are actually coextensive in the TFM simulator. Obviously, this lacks a good degree of physical realism since the proteins being represented are unlikely to be able to coexist in such a way. This spatial coexistence was explicitly prohibited in the simulator of [17], by introduction of a mutually repulsive force between any two monomers. This repulsive force was necessary to produce spirally malformed capsids. I did not attempt to include such a repulsive force in the TFM simulator, because of possible problems with computational efficiency and because the study of spiral malformations was not my primary goal. But it seems reasonable to believe that inclusion of such a "mutual exclusion" force, applied to topologies such as that shown in figure 6.8, would result in the well-known spiral malformation.

Proportions of Various Structures Formed

I have done few experimental runs to determine how often perfect capsids are formed by the simulator, so any results on the proportions of various types of structures formed must be regarded as preliminary. However, the one large run I have made to date has results as follows:

The simulator was run 64 times, with an unoptimized rule set known to produce $T = 3$ capsids sometimes (this was the rule set used to produce figures 6.6 and 6.7).

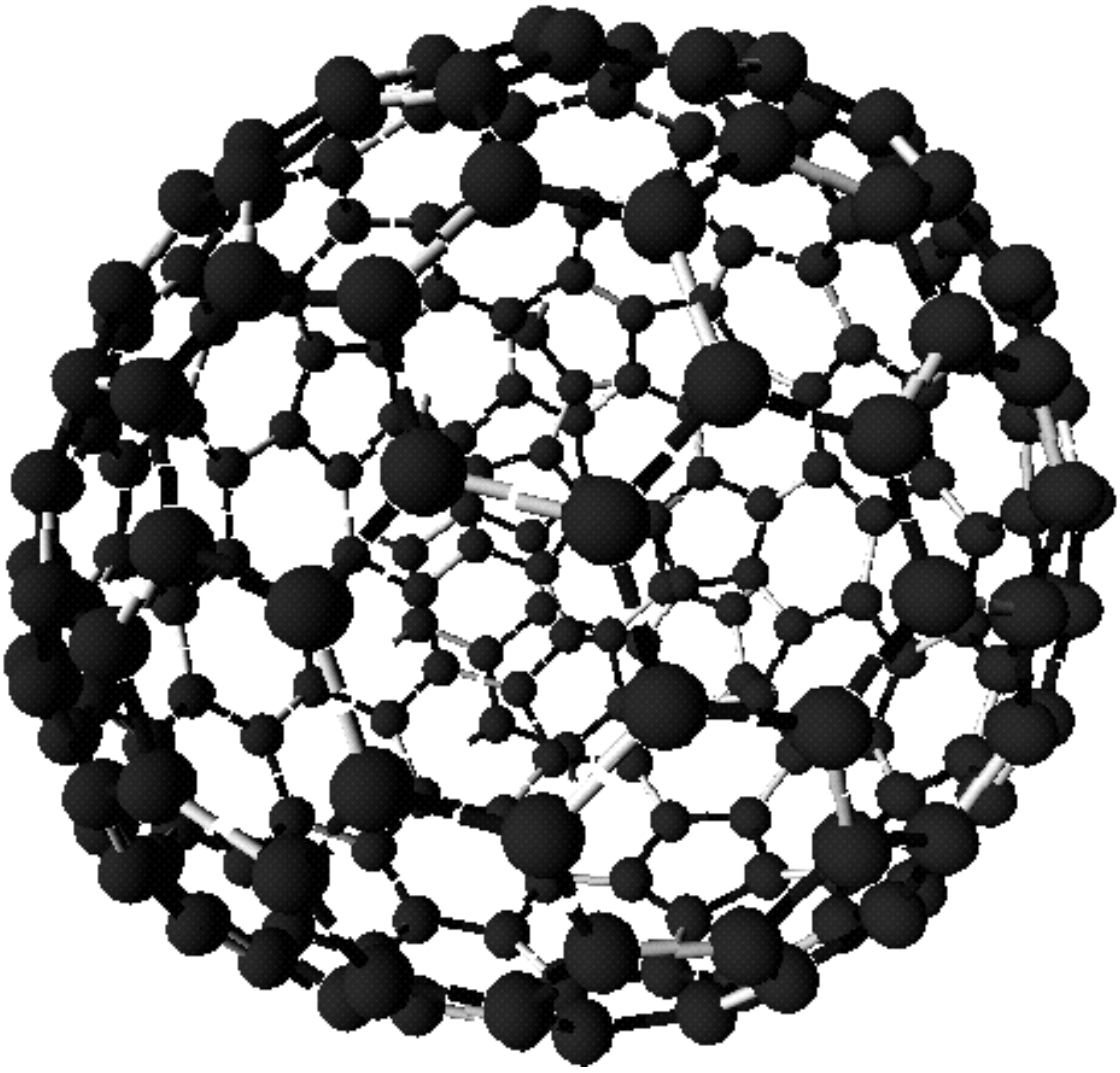

Figure 6.8: When multiple "leaves" of a nascent capsid collide and cannot correctly bind, the result can be a "spiral malformation". Notice the vertical "fissure" of spatially overlapping monomers toward the back of this structure.

Each simulation consisted of 5000 time steps at high temperature, followed by a "fill-in" phase, where any empty binding sites remaining in the structure were filled in without backtracking.

Of the 64 structures formed, 15 were malformed, 23 were odd size capsids, 17 were semiperfect, and 9 were perfect. These results are very encouraging indeed, given that this rule set was hand-optimized (and not very well, at that). A summary is shown in the table below.

type	number	percentage
perfect	9	14%
semi-perfect	17	27%
odd size	23	36%
malformed	15	23%

6.2 The TFM Suggests an Evolutionary Pathway from Small Viruses to Large

It has been noted by Berger and Shor [2], that the monomer geometries required to generate $T = 4$ and $T = 7$ capsids are remarkably close. In fact, they observed in this case that $T = 4$ and $T = 7$ capsids could both be formed using the *same* set of local rules, with the exception of a single disambiguating rule to be applied only once per icosahedral face (see section 1.2). Further, it has been noted in the laboratory that $T = 4$ and $T = 7$ capsids can be formed by the *same* monomer, in varying relative concentrations depending on the ambient chemical environment [14, 12].

Clearly, then, $T = 4$ and $T = 7$ capsids are closely related in their monomer geometries. This, combined with the restrictions on monomer geometry discussed above, together suggest the question of whether higher T-number capsids could have evolved from lower T-numbers. For instance, a virus with a $T = 4$ capsid could have at one point experienced a mutation which increased the IN – OUT angle's natural value. This would have resulted in a lower pentamer density, and a larger implicit

capsid radius at the same time. This, in turn, could have resulted in $T = 7$ capsids forming instead of, or in addition to, $T = 4$ capsids. If this mutation were to survive long enough in the virus population, it could conceivably confer a selective advantage to those viruses possessing it by allowing the coevolution of a larger, hence possibly more sophisticated viral genome.

This process could be repeated several times, each time resulting in larger and larger capsids. Of course, this is completely speculative, but one could evaluate the feasibility of this evolutionary mechanism by enhancing the simulator to support simulated evolution (see section 6.3.3 below).

6.3 Conclusions

6.3.1 Virology

I believe some of the most important conclusions allowed by this work which are specifically pertinent to the field of virology are:

- Cumulative strain models are a simple and believable mechanism for introducing global symmetry into a locally interacting system. In particular, the TFM may explain the emergence of globally symmetric structure in small to medium sized quasi-equivalent virus capsids.

- The TFM also demonstrates the possibility of semi-perfect structures, and gives us a reasonable hypothesis as to how and why they might actually form *in vivo*. Since slightly malformed capsids are observed quite frequently *in vivo*, it is plausible that they may be of the semi-perfect variety, and this question certainly deserves closer scrutiny in the laboratory.

- The TFM gives a possible explanation to hexameric skew *in vivo*, by illustrating that, within the context of the TFM, hexameric skew is essential to symmetric pentamer placement.

- The TFM can model experimentally observed malformations, such as the spiraling malformation.

- The TFM suggests a possible evolutionary pathway from small (low T-number) viruses to large (high T-number).

In addition, I find it worth noting that without the increased flexibility, configurability, and fine-grained control afforded by the TFM simulator, I would not have made so many interesting observations or raised so many new questions. The simulator has definitely been a source of new problems to think about, and new ideas. In this sense as much as any of the above, I consider this work to be a success so far.

6.3.2 Protein-Protein Complexes and Emergent Structure

I would also like to make some concluding remarks about the structure of protein-protein complexes in general.

Perhaps the most striking feature of the TFM, and its greatest contribution to my personal understanding, has been the fact that pairwise interactions of an "only slightly more than trivial" nature, (*i.e.*, different spring constants for flexure in different directions) can describe structures surprisingly rich in emergent detail and symmetry, and that these emergent features can be difficult to predict ahead of time just by considering the structure of the interaction rule. In this context, simulation becomes all the more useful and relevant a tool for the study of this type of emergent structure.

Also, it is apparent that proteins can be used to make structures of such type, more effectively than atoms or simpler molecules, for several reasons:

- The shape of proteins can be almost arbitrarily varied: Unlike atoms, the interaction sites of proteins can be placed in almost arbitrary positions and orientations relative to each other.

- The structure of a protein may be flexible, and further, such flexibility may easily be anisotropic.

- Protein-protein interactions can be extremely specific to the types of proteins involved; more specific than atomic interactions. Also, the number of types of proteins is vastly larger than the number of types of atoms, so the combinatorial possibilities for protein-protein interactions is correspondingly richer.

- Inter-protein interactions can be anisotropic, with the energy of interaction depending sharply on relative orientation; whereas in the case of individual atoms, there is little angle dependence.

The above sources of richness in interaction conspire to give proteins the singular ability to assemble in complex ways, realizing substantial and often surprising emergent structural properties from only local interactions. The study of types of specific interactions, and a taxonomy of emergent structures produced by them, is a wide open area for further research.

6.3.3 Future Work

Besides the obvious future objectives of forming larger and more complex capsids with the TFM, attempting to duplicate with greater accuracy the malformations found in nature, and attempting to demonstrate an evolutionary pathway from small viruses to large, several other refinements and pathways suggest themselves:

Simulator Enhancements

The current simulator is severely restricted at this point because it is single threaded. A marked improvement would be to parallelize the optimization step, as has been done, for example, by Schwartz *et al.* [23, 24].

Another significant speed improvement might be to alter the candidate distribution Q (see equation 5.3) to favor adding nodes rather than deleting them. Currently, when the capsid is almost complete, most candidate moves which are selected as discussed in section 5.3.2 are node deletions, since there are few open binding sites. So the simulator essentially "thrashes" for long periods of time, repeatedly picking a candidate node to delete, determining that doing so will not improve the energy state, and so (usually) rejecting the move. It is typically a long time between steps in which the simulator actually finds an empty binding site and so chooses to try adding a node.

Changing Q to favor picking empty binding sites should fix this speed problem without changing the equilibrium distribution, as long as we are careful to explicitly include the term shown in equation 5.10, which has been omitted to this point for simplicity. Computing this term is not difficult, so including it in the simulation should be a simple matter.

Dynamics

Numerical Scaling and Units It should be noted that all the numerical values used in this work are expressed in arbitrary units. An obvious enhancement would be to scale all the units correctly, as a sanity check against natural capsid growth. For instance, calculating the simulation time step would be a first cut at deciding whether this model might be a reasonable explanation for how things actually happen.

Energy Parameters and Their Effect on Form and Growth Rate When doing the simulations for this thesis, I used $E_{gibbs} < E_{bond} < 0$, to drive the simulation forward from the beginning. However, there are two problems with this approach:

- The monomers are perfectly happy sticking stably onto the structure with only one bond, since doing so is energetically favorable. This results in spiny, dendritic structures growing which often are the wrong shape to bind with each

other. This in turn forces me to run the simulation at unreasonably high temperatures, to introduce enough backtracking to allow capsids to properly initiate. It is believed that such spiny dendritic structures do not occur in nature, which would indicate that this set of simulation parameters is not physically reasonable.

- Structures nucleate very quickly and with high probability. This is good for starting the simulation, but is chemically unbelievable in the following sense: If capsids were so quick and common to nucleate, they would frequently nucleate next to each other in such close proximity as to interfere with each other's growth. This would result in malformed structures formed of multiple incomplete capsids, fused together. Such structures are indeed found in nature, and have been effectively simulated by Schwartz [23].

It is therefore reasonable to expect that capsids in nature nucleate rarely. Further, it is believed that capsids grow almost sequentially, once they nucleate.

In order to achieve these conditions in the TFM simulator, we could run simulations with $E_{gibbs} > 0$ and $E_{bond} << 0$. This would result in probabilistic nucleation, followed by fast sequential growth. Moreover, the sequential growth would be dominated by node additions which form *two* bonds instead of just one, thus effectively eliminating the problem of dendritic growth.

This change would therefore enhance the realism of the simulation in two ways:

- Increasing realism during the growth phase by causing more nearly sequential assembly and by favoring more bonds.

- Increasing the believability of the "single structure" assumption by engendering a low probability of nucleation.

Medicine / Therapeutics

Let us assume we have assigned physically reasonable units to all the simulation parameters, as described in the last section. We may now take up the question of modeling the exact response of capsids to various physical stimuli. For instance, if we have a reasonable guess as to the mass of the monomers, their size, and the flexibility parameters, we could compute the vibrational resonant modes of the capsid, along with resonant Q, *etc.* We might be able to compare the simulation results so obtained with spectroscopic data from laboratory experiments, to further evaluate the accuracy of this model and to fine tune the parameters.

Data on vibrational frequencies, modes, and damping factors might also have medical uses. If it is found that some capsids have a resonant Q high enough to allow their destruction with low power density, narrow band emissions, then the therapeutic implications could be profound.

Materials Science

As mentioned above in section 6.3.2, the variety and richness of structures producible by protein-protein interaction is truly astounding. It has not escaped my attention that, assuming we know at some point soon how to make "designer proteins" to specific shape and binding specifications, that an entire new area of materials science opens up to us: The science of complex mechano-chemical self-assembling materials.

I believe that the selective bonding/selective flexibility approach to protein-protein interaction modeling, coupled with efficient and easy to use simulators, may be advantageously used in the design of such materials. Such an approach is being used, for instance, by Hobbs *et al.* [9, 11, 8], to model topologically disordered ceramic structures.

Some interesting starter problems in this field might be the design of auto-assembling toroidal vesicles, or more ambitiously, the design of percolating pseudorandom spongiform materials. Such materials would be composed of large areas of orientable surface,

reticulated into a dense three dimensional (space filling) structure. Ideally, the fractal dimension of the surface should be controllable by varying some parameter of the assembling protein subunits, as would the ratio of included to excluded volumes and the degree of percolation. Such materials could be used to great advantage in many filtering tasks and continuous-flow reaction problems, for example.

Shape changing gels and other novel materials might also be possible based on this approach; it is certainly a field for further work.

Bibliography

[1] B. Berger, R. Schwartz, P.W. Shor, and P.E. Prevelige, Jr. Kinetic modeling of virus capsid assembly. In *Biophys. J.*, volume 70, page A364, February 1996.

[2] Bonnie Berger and Peter W. Shor. Local rule switching mechanism for viral shell geometry. In *Proc. 14th Biennial Conference on Phage/Virus Assembly*, June 1995. Full version appears in MIT-LCS TM #527.

[3] Bonnie Berger and Peter W. Shor. On the mathematics of virus shell assembly. Technical Report TM-519, MIT Lab. for Computer Science, April 1995.

[4] Bonnie Berger, Peter W. Shor, Lisa Tucker-Kellogg, and J. King. Local rule-based theory of virus shell assembly. *Proc. of the Natl. Academy of Sci.*, 91(16), August 1994.

[5] D. L. D. Caspar and A. Klug. Physical principles in the construction of regular viruses. *Cold Spring Harbor Symp. Quant. Biol.*, 27:1–24, 1962.

[6] W. Earnshaw and J. King. Structure of phage P22 coat protein aggregates formed in the absence of the scaffolding protein. *J. Mol. Biol.*, 126:721–747, 1978.

[7] R. W. Hendrix. Shape determination in virus assembly: The bacteriophage example. In S. Casjens, editor, *Virus Structure and Assembly*, pages 169–204. Jones and Bartlett, Boston, MA, 1985.

[8] L.W. Hobbs, C.E. Jesurum, V. Pulim, and B.A. Berger. Local topology of silica networks. *Philosophical Magazine*, 1998. In press.

[9] L.W. Hobbs, A.N. Sreeram, C.E. Jesurum, and B. Berger. Structural freedom, topological disorder, and the irradiation-induced amorphization of ceramic structures. *Nuclear Instruments and Methods in Physical Research B*, 116:18–25, 1996.

[10] Mark Jerrum and Alistair Sinclair. The markov chain monte carlo method: An approach to approximate counting and integration. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, chapter 12, pages 482–520. PWS Publishing, Boston, MA, 1996.

[11] C.E. Jesurum, V. Pulim, L.W. Hobbs, and B. Berger. Modeling of topologically disordered tetrahedral network structures using local rules. In *Proc. of the 13th International Conference on Defects in Insulating Materials*, pages 37–40. Trans Tech Publications, July 1996.

[12] I. Katsura. Structure and inherent properties of the bacteriophage lambda head shell IV: Small-head mutants. *J. Mol. Biol.*, 171:297–317, 1983.

[13] J. King, R. Griffin-Shea, and M. T. Fuller. Scaffolding proteins and the genetic control of virus shell assembly. *Quart. Rev. Biol.*, 55:369–393, 1980.

[14] Ole J. Marvik, Praveen Sharma, Terje Dokland, and Bjorn H. Lindqvist. Bacteriophage P2 and P4 assembly: alternate scaffolding proteins regulate capsid size. *Virology*, 200:702–714, 1994.

[15] Christopher J. Marzec and Loren A. Day. Pattern formation in icosahedral virus capsids: The papova viruses and nudaurelia capensis beta virus. *Biophys. J.*, 65:2559–2577, 1993.

[16] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[17] D. Muir. Simulated construction of viral protein shells. Bachelor's Thesis. Massachusetts Institute of Technology, 1994.

[18] Radford M. Neal. Probabilistic inference using markov chain monte carlo methods. Technical Report CRG-TR-93-1, University of Toronto, Department of Computer Science, University of Toronto, September 1993.

[19] John K. Ousterhout. *Tcl and the Tk Toolkit.* Addison-Wesley, Reading, Massachusetts, 1997.

[20] Michael E. Pique. Rotation tools. In Andrew S. Glassner, editor, *Graphics Gems*, chapter 9, pages 465–469. AP Professional, Chestnut Hill, MA, 1990.

[21] P. Prevelige, A. Zlotnick, P. Thuman-Commike, and David Coombs. Personal communication.

[22] Eric Raible. Matrix orthogonalization. In Andrew S. Glassner, editor, *Graphics Gems*, chapter 9, page 464. AP Professional, Chestnut Hill, MA, 1990.

[23] R.S. Schwartz. A multi-threaded simulator for the kinetics of virus shell assembly. Master of Engineering Thesis. Massachusetts Institute of Technology, June 1996.

[24] Russell S. Schwartz, Peter E. Prevelige, Peter W. Shor, and Bonnie A. Berger. Local rules simulation of virus capsid assembly. unpublished manuscript, August 1997.

[25] Tibor Tarnai, Zsolt Gaspar, and Lidia Szalai. Pentagon packing models for 'all-pentamer' virus structures. *Biophys. J.*, 69:612–618, 1995.

[26] P. A. Thuman-Commike, B. Greene, J. Jakana, B. V. V. Prasad, J. King, P. E. Prevelige, Jr., and W. Chiu. Three-dimensional structure of scaffolding-containing phage P22 procapsids by electron cryo-microscopy. *J. Mol. Biol.*, 260:85–98, 1996.